State-dependent importance sampling schemes via minimum cross-entropy

Ridder, A.A.N.; Taimre, T.

published in
Annals of Operations Research
2011

DOI (link to publisher)
10.1007/s10479-009-0611-7

document version
Publisher's PDF, also known as Version of record

Ann Oper Res (2011) 189:357–388
DOI 10.1007/s10479-009-0611-7

State-dependent importance sampling schemes via minimum cross-entropy

Ad Ridder · Thomas Taimre

Published online: 27 August 2009

Abstract We present a method to obtain state- and time-dependent importance sampling estimators by repeatedly solving a minimum cross-entropy (MCE) program as the simulation progresses. This MCE-based approach lends a foundation to the natural notion to stop changing the measure when it is no longer needed. We use this method to obtain a state- and time-dependent estimator for the one-tailed probability of a light-tailed i.i.d. sum that is logarithmically efficient in general and strongly efficient when the jumps are Gaussian. We go on to construct an estimator for the two-tailed problem which is shown to be similarly efficient. We consider minor variants of the algorithm obtained via MCE, and present some numerical comparisons between our algorithms and others from the literature.

Keywords Cross-entropy · Rare events · Importance sampling · State dependence

1 Introduction

Let $\boldsymbol{X}(n) = (X_1, X_2, \ldots, X_n)$ $(n \in \mathbb{N})$ be a vector of random variables with joint probability density function $f(\boldsymbol{x})$, where we allow both continuous and discrete models. Consider an event $A(n)$ in the σ-algebra over the sample space of $\boldsymbol{X}(n)$. We are interested in estimating the probability

$$\ell(n) = P(\boldsymbol{X}(n) \in A(n)),$$

assuming that $\ell(n) \to 0$ as we let $n \to \infty$. Hence, we say that $A(n)$ is a rare event when the rarity parameter n becomes large. In this paper we study an importance sampling algorithm

The research of T. Taimre supported by the Commonwealth Government of Australia, and by the Australian Research Council Centre of Excellence for Mathematics and Statistics of Complex Systems.

A. Ridder (✉)
Department of Econometrics and Operations Research, Vrije University, Amsterdam, Netherlands
e-mail: aridder@feweb.vu.nl

T. Taimre
Department of Mathematics, The University of Queensland, Brisbane, Australia
e-mail: ttaimre@maths.uq.edu.au

for estimating the rare-event probability $\ell(n)$ by simulation with a new (or importance sampling) density $g(\boldsymbol{x})$. The algorithm is based on minimising the Kullback-Leibler divergence of the original density $f(\boldsymbol{x})$ from a family of densities $g(\boldsymbol{x})$, subject to a set of integral constraints. That is, $g(\boldsymbol{x})$ is given by the solution of the following mathematical program:

$$\begin{aligned} &\inf_{g\geq 0} \mathcal{D}_{\mathrm{KL}}(g|f) \\ &\text{s.t.} \quad \int g(\boldsymbol{x})\,d\boldsymbol{x} = 1, \\ &\qquad E_g[c_i(\boldsymbol{X}(n))] = b_i, \quad i \in \mathcal{E}, \\ &\qquad E_g[c_i(\boldsymbol{X}(n))] \geq b_i, \quad i \in \mathcal{I}. \end{aligned} \tag{1}$$

This is the generic minimum cross-entropy (MCE) program with the Kullback-Leibler divergence as objective function, which is

$$\mathcal{D}_{\mathrm{KL}}(g|f) = \int g(\boldsymbol{x}) \log \frac{g(\boldsymbol{x})}{f(\boldsymbol{x})}\,d\boldsymbol{x},$$

together with known constraint functions $c_i : \mathbb{R}^n \to \mathbb{R}$ and right-hand sides $b_i \in \mathbb{R}$. We allow these right-hand sides to depend on the parameter n.

The basic MCE idea goes back probably to Jaynes (1957) who considered the problem of maximizing Shannon's entropy under additional constraints, also known as the Maximum Entropy Principle for probability inference. The corresponding optimization program is similar to our program (1) by replacing the objective by maximizing Shannon's entropy

$$\mathcal{D}_{\mathrm{S}}(g) = -\int g(\boldsymbol{x}) \log g(\boldsymbol{x})\,d\boldsymbol{x}.$$

When the support of the function g is bounded, we might as well minimize the Kullback-Leibler divergence of the uniform density u:

$$\sup_g \mathcal{D}_{\mathrm{S}}(g) = \inf_g \mathcal{D}_{\mathrm{KL}}(g|u).$$

Finally, the natural extension is to consider some given density f allowing infinite support. This program has been formulated originally in Jaynes (1963); Kullback and Khairat (1966) under the name of the Principle of Minimum Discrimination Information.

Since then, MCE programs have been studied and used in a wide variety of research fields such as information theory, natural language processing, utility theory, computer vision, spatial physics, statistical mechanics, statistical data analysis, etc. Their usage for rare-event simulation is relatively new and has been reported in Rubinstein (2005); Ridder and Rubinstein (2007); Botev et al. (2007).

Suppose that we use a solution $g(\boldsymbol{x})$ of (1) as the probability density function for generating samples $\boldsymbol{x}$ of the random vector $\boldsymbol{X}(n)$. Then the importance sampling estimator after k samples is

$$Y(n)[k] = \frac{1}{k}\sum_{r=1}^{k} L\left(\boldsymbol{X}^{(r)}(n)\right) 1\left\{\boldsymbol{X}^{(r)}(n) \in A(n)\right\}, \tag{2}$$

where the $\boldsymbol{X}^{(r)}(n)$ are i.i.d. with probability density function $g(\boldsymbol{x})$, and unbiasedness is guaranteed by the likelihood ratio $L(\boldsymbol{x}) = f(\boldsymbol{x})/g(\boldsymbol{x})$. We denote the r-th term in (2) by $Y^{(r)}(n)$, and an arbitrary term by just $Y(n)$ (which is also the single-sample estimator $Y(n)[1]$).

To put these matters into perspective, suppose that the individual members $X_1, X_2, \ldots$ of $\boldsymbol{X}(n)$ are i.i.d., and that

$$A(n) = \left\{ \boldsymbol{x} \in \mathbb{R}^n : \sum_{j=1}^{n} x_j \geq bn \right\} \quad \text{where } b > E_f[X_1].$$

This is the classical rare-event problem concerning tail probabilities of sums of i.i.d. increments or jumps X_j, which has been studied extensively in the simulation literature, see Bucklew (2004) for an overview. Abusing notation slightly, a generic jump X has probability density function $f(x)$ which is assumed here to be light-tailed. Consider the MCE program (1) with a single equality constraint $E_g[\sum_{j=1}^{n} X_j] = bn$ in addition to the normalisation condition $\int g(\boldsymbol{x})\, d\boldsymbol{x} = 1$. Then it is well known (see e.g. Ridder and Rubinstein 2007) that the solution $g(\boldsymbol{x})$ coincides with an exponential change of measure under which all of the jumps remain i.i.d, the probability density function of the individual jumps is given by

$$g(x) = f(x)e^{\lambda x - \psi(\lambda)}, \tag{3}$$

where $\psi(\theta) = \log E_f[\exp(\theta X)]$ is the cumulant generating function of a jump, and the specific tilting parameter λ satisfies $\psi'(\lambda) = b$. We recognise this as the same exponentially tilted solution as is obtained by the large deviations approach by letting $n \to \infty$ (Bucklew 2004).

The importance sampling density for the jumps in this problem given in (3) is fixed throughout the sampling process, irrespective of the state $S_j = X_1 + \cdots + X_j$ after the first j jumps. Such state-independent importance sampling algorithms are known to be inefficient for models with nontrivial rare events (Glassermann and Wang 1997) or for queueing network models with buffer overflow rare events (de Boer 2006). Efficiency may be defined as follows (Heidelberger 1995; Bucklew 2004; L'Ecuyer et al. 2008).

Definition 1 An importance sampling algorithm or its associated importance sampling estimator $Y(n)$ is strongly efficient if it has bounded relative error:

$$\limsup_{n\to\infty} \frac{E[(Y(n))^2]}{(E[Y(n)])^2} < \infty.$$

It is logarithmically efficient (or asymptotically optimal) if

$$\lim_{n\to\infty} \frac{\log E[(Y(n))^2]}{\log E[Y(n)]} = 2. \tag{4}$$

Notice that it suffices to define these efficiencies for the single-sample estimator, because the sample average estimator $Y(n)[k]$ has decreasing variance with constant mean, and thus has decreasing second moment as the sample size k increases.

The contribution of our paper has the following aspects.

- We propose a state- and time-dependent importance sampling algorithm to estimate the one-tailed probability

$$P\left(\sum_{j=1}^{n} X_j \geq bn \right), \tag{5}$$

where $X_1, \ldots, X_n$ are i.i.d. with light-tailed distribution, and where the overflow level $b > E[X]$. It is well known that this probability decays exponentially fast to zero as $n \to \infty$. The importance sampling density function of the $k+1$-th jump is found via an MCE program. The resulting density is either the original density f, or an exponentially tilted version of it, the choice depending on time k and state S_k. We shall prove that the associated estimator is logarithmically efficient in general, and strongly efficient in case of Gaussian jumps. For the latter we relied heavily on the results of Blanchet and Glynn (2006) who constructed a strongly efficient algorithm via another approach, and whose resulting importance sampling densities were almost the same as ours. We were not able to prove strong efficiency in general, but the simulation results for other light-tailed jump distributions seem to indicate that this holds true. A possible explanation is that for sufficiently large n, the states S_n are approximately Gaussian distributed, and behave as though the individual jumps in each sum were Gaussian. We assessed our algorithm and several variations thereof by executing extensive simulation experiments with it and its variants, alongside the traditional state-independent exponentially tilting algorithm (Heidelberger 1995; Bucklew 2004) and the algorithm recently developed by Blanchet and Glynn (2006). We found that our estimator outperforms the traditional one, and is slightly better than the Blanchet-Glynn estimator.

- Subsequently, we consider estimating the two-tailed probability

$$P\left(\sum_{j=1}^{n} X_j \leq an \text{ or } \sum_{j=1}^{n} X_j \geq bn\right), \tag{6}$$

again with i.i.d. jumps, and where $a < E[X] < b$ such that state-independent importance sampling algorithms without mixing are inefficient (Glassermann and Wang 1997). We consider the estimator obtained by mixing two of our one-tailed estimators. Under the condition that the mixing probabilities do not decay to zero exponentially fast, we show logarithmic efficiency for our mixed estimator in general. In the case of Gaussian jumps we again obtain strong efficiency.

The two-tailed problem has been studied before as a typical example where the 'naive' large deviations approach tends to fail logarithmic efficiency. These studies give resolutions as well: Glassermann and Wang (1997) propose a mixed importance sampling estimator; also Bucklew (2004) in Example 5.2.13 involving Gaussian jumps comes up with the same mixed estimator; Dupuis and Wang (2004) pursue another approach to construct a time- and state-dependent importance sampling algorithm based on the solution of an Isaacs equation; Dupuis and Wang (2007) give an algorithm based on a subsolution of an Isaacs equation. These three algorithms are proven to be logarithmically efficient. Our algorithm is in the line of the first one but differs in the definition of the one-tailed estimators of the mixture. Furthermore, we let the mixing probabilities to be determined by an appropriate MCE program. We compare our estimator with these three other algorithms, and we find a large improvement over Glassermann and Wang (1997); Dupuis and Wang (2007), and a small improvement over Dupuis and Wang (2004).

One may object against the two-tailed problem that—as we will see in Sect. 2.2—it 'just' needs efficient estimators for the two parts (one-tailed problems) and the right mixing probabilities. There are many interesting problems in which the rare event cannot be decomposed in disjoint 'easy' problems. For instance, consider Jackson networks with at least two queues. The set of states where at least one of the queues exceeds the level n (and $n \to \infty$)

is such a rare event. It is challenging to investigate these problems in relation to our MCE importance sampling, but in our opinion, this falls outside the scope of our paper.

The paper is organised as follows. Section 2 gives the solution to the MCE program (1), and describes three ways of defining mixed importance sampling estimators, along with conditions for efficiency of these from their component estimators. In Sect. 3 we present our algorithm for the one-tailed problem (5), we prove logarithmic efficiency in general and strong efficiency in the case of Gaussian jumps, and we show simulation results. In Sect. 4 we do the same for the two-tailed problem (6), and we conclude with a few final remarks in Sect. 5.

2 Preliminaries

2.1 Solving the minimum cross-entropy program

The Kullback-Leibler MCE program (1) is solved by applying the method of Lagrange multipliers. The solution is nonnegative and is given by Rubinstein and Kroese (2008, Sect. 9.5)

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) \exp\left(\lambda_0 + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\boldsymbol{x})\right),$$

where the λ_i's solve the dual program

$$\sup_{\lambda_0, \lambda_i} \lambda_0 + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i b_i - e^{\lambda_0} E_f \left[\exp\left(\sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\boldsymbol{X}(n)) \right) \right],$$

subject to the restriction that $\lambda_i \geq 0$ for the inequality constraints $i \in \mathcal{I}$.

2.2 Efficiency of mixed importance sampling estimators

In this section we analyse the efficiency of a mixed importance sampling estimator in general terms. For that purpose, we suppose that the rare event $A(n)$ is partitioned into m disjoint subsets

$$A(n) = \bigcup_{j=1}^{m} A_j(n),$$

such that $P(A_j(n)) > 0$ for all j and n, ensuring that $\sum_{j=1}^{m} P(A_j(n)) = P(A(n))$. Furthermore we assume that there are unbiased importance sampling estimators of the probabilities $P(A_j(n))$ with associated importance sampling density functions $g_j(\boldsymbol{x})$, likelihood ratios $L_j(\boldsymbol{x}) = f(\boldsymbol{x})/g_j(\boldsymbol{x})$, and corresponding single-sample estimators given by

$$Y_j(n) = L_j(\boldsymbol{X}(n)) 1\left\{\boldsymbol{X}(n) \in A_j(n)\right\}.$$

For our purposes, a mixed importance sampling estimator for $P(A(n))$ mixes the individual estimators in either random or deterministic proportions. First we consider the random version.

Definition 2 For any n, let $\Delta(n)$ be a random variable on $\{1, 2, \ldots, m\}$ with positive probabilities $p_j(n) > 0$, $\sum_{j=1}^{m} p_j(n) = 1$, which may depend on n, but such that $\Delta(n)$ is independent of the $Y_j(n)$'s. Then the mixed importance sampling estimator is defined by

$$Y(n) = \sum_{j=1}^{m} \frac{1}{p_j(n)} 1\{\Delta(n) = j\} Y_j(n). \tag{7}$$

When we substitute the individual estimators into (7), we get

$$\begin{aligned} Y(n) &= \sum_{j=1}^{m} \frac{1}{p_j(n)} 1\{\Delta(n) = j\} L_j(\boldsymbol{X}(n)) 1\left\{\boldsymbol{X}(n) \in A_j(n)\right\} \\ &= \sum_{j=1}^{m} 1\{\Delta(n) = j\} \frac{f(\boldsymbol{X}(n))}{p_j(n) g_j(\boldsymbol{X}(n))} 1\left\{\boldsymbol{X}(n) \in A_j(n)\right\}. \end{aligned}$$

From this, we see how $Y(n)$ is implemented: realise $\Delta(n)$, and depending on its outcome realise $\boldsymbol{X}(n)$ according to density $g_{\Delta(n)}$. Finally, check whether $\boldsymbol{X}(n) \in A_{\Delta(n)}(n)$ (activating the corresponding indicator). As a consequence of the relations

$$E\left[1\{\Delta(n) = j\} Y_j(n)\right] = E\left[1\{\Delta(n) = j\}\right] E\left[Y_j(n)\right] = p_j(n) P\left(A_j(n)\right),$$

the mixed estimator is unbiased:

$$\begin{aligned} E[Y(n)] &= E\left[\sum_{j=1}^{m} \frac{1}{p_j(n)} 1\{\Delta(n) = j\} Y_j(n)\right] = \sum_{j=1}^{m} \frac{1}{p_j(n)} E\left[1\{\Delta(n) = j\} Y_j(n)\right] \\ &= \sum_{j=1}^{m} P\left(A_j(n)\right) = P(A(n)). \end{aligned}$$

Our goal is to show that under certain conditions the mixed importance sampling estimator $Y(n)$ is strongly or logarithmically efficient when its individual members $Y_j(n)$ are similarly efficient. Though this seems natural, it is not trivial. Further, we could not find references, except for special cases of mixing exponentially tilted importance sampling densities, for instance Sadowsky and Bucklew (1990) and Glassermann and Wang (1997). Therefore we shall give sufficient conditions the mixed estimator to inherit efficiency from its component estimators, and prove the given results. To proceed, we need the second moment and squared first moment of the mixed importance sampling estimator in terms of the corresponding component quantities. For the second moment, we have that

$$\begin{aligned} E[Y^2(n)] &= E\left[\left(\sum_{j=1}^{m} \frac{1}{p_j(n)} 1\{\Delta(n) = j\} Y_j(n)\right)^2\right] \\ &= E\left[\sum_{j=1}^{m} \frac{1}{p_j(n)^2} 1\{\Delta(n) = j\} Y_j^2(n)\right. \\ &\quad \left. + \sum_{i \neq j} \frac{1}{p_i(n)} \frac{1}{p_j(n)} \underbrace{1\{\Delta(n) = i\} 1\{\Delta(n) = j\}}_{=0} Y_i(n) Y_j(n)\right] \end{aligned}$$

$$= \sum_{j=1}^{m} \frac{1}{p_j(n)^2} E[1\{\Delta(n) = j\} Y_j^2(n)]$$

$$= \sum_{j=1}^{m} \frac{1}{p_j(n)} E[Y_j^2(n)].$$

Since all terms are positive, for the squared first moment we have

$$\begin{aligned}(E[Y(n)])^2 &= \left(\sum_{j=1}^{m} \frac{1}{p_j(n)} E[1\{\Delta(n) = j\} Y_j(n)] \right)^2 \\ &\geq \sum_{j=1}^{m} \left(\frac{1}{p_j(n)} E[1\{\Delta(n) = j\} Y_j(n)] \right)^2 \\ &= \sum_{j=1}^{m} (E[Y_j(n)])^2.\end{aligned}$$

Strong efficiency of the mixed estimator is obtained easily when all its individual members are strongly efficient.

Lemma 1 *Assume that there are finite constants c_j $(j = 1, \ldots, m)$ s.t.*

$$\limsup_{n\to\infty} \frac{E[Y_j^2(n)]}{(E[Y_j(n)])^2} \leq c_j.$$

Then the mixed estimator is strongly efficient.

Proof Firstly, apply the findings of the squared first moment, and the second moment to obtain the inequalities

$$\frac{E[Y^2(n)]}{(E[Y(n)])^2} \leq \frac{\sum_{j=1}^{m} \frac{1}{p_j(n)} E[Y_j^2(n)]}{\sum_{j=1}^{m} (E[Y_j(n)])^2} \leq \sum_{j=1}^{m} \frac{1}{p_j(n)} \frac{E[Y_j^2(n)]}{(E[Y_j(n)])^2}.$$

Then it follows that

$$\begin{aligned}\limsup_{n\to\infty} \frac{E[Y^2(n)]}{(E[Y(n)])^2} &\leq \limsup_{n\to\infty} \sum_{j=1}^{m} \frac{1}{p_j(n)} \frac{E[Y_j^2(n)]}{(E[Y_j(n)])^2} \\ &\leq \sum_{j=1}^{m} \frac{1}{p_j(n)} \limsup_{n\to\infty} \frac{E[Y_j^2(n)]}{(E[Y_j(n)])^2} \\ &\leq \sum_{j=1}^{m} \frac{1}{p_j(n)} c_j < \infty.\end{aligned}$$

□

It is more involved to obtain logarithmic efficiency of the mixed estimator from corresponding logarithmic efficiencies of its individual members.

Assumption 1 *For any j*:

(a) *the sequence of probabilities $(P(A_j(n)))_n$ satisfy a large deviations limit*:

$$\lim_{n\to\infty} \frac{1}{n} \log P(A_j(n)) = -I_j,$$

for some $0 < I_j < \infty$;

(b) *the estimator $Y_j(n)$ is logarithmically efficient*:

$$\lim_{n\to\infty} \frac{\log E[(Y_j(n))^2]}{\log E[Y_j(n)]} = 2;$$

(c) *the sequence of mixing probabilities $(p_j(n))_n$ may not tend to zero exponentially fast*:

$$\lim_{n\to\infty} \frac{1}{n} \log \frac{1}{p_j(n)} = 0.$$

Lemma 2 *Under the conditions of Assumption 1, the mixed importance sampling estimator $Y(n)$ is logarithmically efficient.*

Proof Firstly, from conditions (a) and (b) of Assumption 1 it follows directly that the first and second moment of the estimators $Y_j(n)$ satisfy the following large deviations limits:

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} \log E\left[Y_j(n)\right] &= -I_j, \\ \lim_{n\to\infty} \frac{1}{n} \log E\left[\left(Y_j(n)\right)^2\right] &= -2I_j. \end{aligned} \tag{8}$$

Since $E[Y(n)] = \sum_{j=1}^m E[Y_j(n)]$ we obtain a large deviations limit for $E[Y(n)]$ by applying the principle of the largest term (Dembo and Zeitouni 1998, Lemma 1.2.15):

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} \log E[Y(n)] &= \lim_{n\to\infty} \frac{1}{n} \log \sum_{j=1}^m E\left[Y_j(n)\right] \\ &= \max_{j=1,\ldots,m} \lim_{n\to\infty} \frac{1}{n} \log E\left[Y_j(n)\right] \\ &= -\min_{j=1,\ldots,m} I_j \doteq -I. \end{aligned}$$

Next, we establish a large deviations limit for $E[(Y(n))^2]$ by considering lower and upper bounds. The lower bound is easily found by applying Jensen's inequality:

$$\begin{aligned} \liminf_{n\to\infty} \frac{1}{n} \log E\left[(Y(n))^2\right] &\geq \liminf_{n\to\infty} \frac{1}{n} \log(E[Y(n)])^2 \\ &= 2 \liminf_{n\to\infty} \frac{1}{n} \log E[Y(n)] = -2I. \end{aligned}$$

For the upper bound we reason as follows:

$$
\begin{aligned}
&\limsup_{n\to\infty}\frac{1}{n}\log E\left[(Y(n))^2\right]\\
&\quad=\limsup_{n\to\infty}\frac{1}{n}\log\sum_{j=1}^{m}\frac{1}{p_j(n)}E\left[\left(Y_j(n)\right)^2\right]\\
&\quad\le\limsup_{n\to\infty}\frac{1}{n}\log m\max_{j=1,\ldots,m}\frac{1}{p_j(n)}E\left[\left(Y_j(n)\right)^2\right]\\
&\quad=\limsup_{n\to\infty}\frac{1}{n}\log\max_{j=1,\ldots,m}\frac{1}{p_j(n)}E\left[\left(Y_j(n)\right)^2\right]\\
&\quad=\limsup_{n\to\infty}\max_{j=1,\ldots,m}\frac{1}{n}\log\frac{1}{p_j(n)}E\left[\left(Y_j(n)\right)^2\right]\\
&\quad\overset{(a)}{=}\max_{j=1,\ldots,m}\limsup_{n\to\infty}\frac{1}{n}\log\frac{1}{p_j(n)}E\left[\left(Y_j(n)\right)^2\right]\\
&\quad\le\max_{j=1,\ldots,m}\left(\limsup_{n\to\infty}\frac{1}{n}\log\frac{1}{p_j(n)}+\limsup_{n\to\infty}\frac{1}{n}\log E\left[\left(Y_j(n)\right)^2\right]\right)\\
&\quad\overset{(b)}{=}\max_{j=1,\ldots,m}\limsup_{n\to\infty}\frac{1}{n}\log E\left[\left(Y_j(n)\right)^2\right]\\
&\quad\overset{(c)}{=}-2\min_{j=1,\ldots,m}I_j=-2I.
\end{aligned}
$$

In (a) we used that the maximum is taken from a finite set (see Appendix); (b) follows from Assumption 1(c); (c) is due to (8). Finally we obtain logarithmic efficiency by noting that

$$
\begin{aligned}
\lim_{n\to\infty}\frac{\log E[(Y(n))^2]}{\log E[Y(n)]}&=\lim_{n\to\infty}\frac{\frac{1}{n}\log E[(Y(n))^2]}{\frac{1}{n}\log E[Y(n)]}\\
&=\frac{\lim_{n\to\infty}\frac{1}{n}\log E[(Y(n))^2]}{\lim_{n\to\infty}\frac{1}{n}\log E[Y(n)]}=\frac{-2I}{-I}=2.
\end{aligned}
$$

□

Remark 1 An alternative approach is to define an importance sampling scheme by mixing deterministic fractions of (independent) estimators as has been pursued in Glassermann and Wang (1997). In this method an overall sample size of k samples is split by allocating a fraction $p_j(n)$ for estimating each $P(A_j(n))$ $(j=1,\ldots,m)$. Such a mixture scheme might be viewed as the 'deterministic' version of the 'randomised' mixing scheme that we have just considered. Thus, when we let $k_j=[p_j(n)k]$ be the number of replications of $Y_j(n)$, the associated 'deterministic' estimator is

$$
Y^{\mathrm{d}}(n)[k]=\sum_{j=1}^{m}\frac{1}{k_j}\sum_{i=1}^{k_j}Y_j^{(i)}(n),
$$

whereas the corresponding 'randomised' estimator is

$$
Y^{\mathrm{r}}(n)[k]=\frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{m}\frac{1}{p_j(n)}1\left\{\Delta^{(i)}(n)=j\right\}Y_j^{(i)}(n).
$$

Clearly, the expected number of replications of $Y_j(n)$ under the 'randomised' scheme equals the same number $kp_j(n)$ of replications used in the 'deterministic' scheme. However, it is well known, and can be easily checked, that

$$\mathrm{Var}\left[Y^{\mathrm{d}}(n)[k]\right] \leq \mathrm{Var}\left[Y^{\mathrm{r}}(n)[k]\right].$$

For this reason we have implemented the 'deterministic' mixed importance sampling estimator in our later numerical experiments.

When the mixing fractions are constant (independent of n), a proof of logarithmic efficiency based on conditions (a) and (b) is given in Glassermann and Wang (1997). They remark that efficiency remains by allocating an asymptotically negligible fraction of samples to events $A_j(n)$ that are less likely, i.e., have large deviations rates $I_j > \min_{t=1,\ldots,m} I_t$. This is in fact our additional condition (c).

Remark 2 A third way of defining a mixed importance sampling estimator would be to perform standard importance sampling using a mixed probability density

$$g(\boldsymbol{x}) = \sum_{j=1}^{m} p_j(n) g_j(\boldsymbol{x}),$$

for some set of mixing probabilities $(p_j(n))_j$. Thus, the associated (single sample) estimator is given by

$$Y^{\mathrm{mix}}(n) = \frac{f(\boldsymbol{X}(n))}{g(\boldsymbol{X}(n))} 1\{\boldsymbol{X}(n) \in A(n)\}. \tag{9}$$

An important feature of this approach is that the subsets $A_j(n)$ do not have to be disjoint, as long as their union forms the rare event $A(n)$. The estimator is clearly unbiased, and analysing its second moment we see that

$$\begin{aligned}
E_g\left[(Y^{\mathrm{mix}}(n))^2\right] &= E_f\left[Y^{\mathrm{mix}}(n)\right] \\
&= E_f\left[\frac{f(\boldsymbol{X}(n))}{g(\boldsymbol{X}(n))} 1\{\boldsymbol{X}(n) \in A(n)\}\right] \\
&\leq \sum_{j=1}^{m} E_f\left[\frac{f(\boldsymbol{X}(n))}{g(\boldsymbol{X}(n))} 1\left\{\boldsymbol{X}(n) \in A_j(n)\right\}\right] \\
&\leq \sum_{j=1}^{m} E_f\left[\frac{f(\boldsymbol{X}(n))}{p_j(n) g_j(\boldsymbol{X}(n))} 1\left\{\boldsymbol{X}(n) \in A_j(n)\right\}\right] \\
&= \sum_{j=1}^{m} \frac{1}{p_j(n)} E_{g_j}\left[(Y_j(n))^2\right] \\
&= E\left[(Y(n))^2\right],
\end{aligned}$$

where $Y(n)$ is the mixed importance sampling estimator previously given in Definition 2. However, because we took into account the computational time for each estimator, we decided to implement the mixture scheme with the deterministic fractions (which also gives a variance reduction over $Y(n)$).

Notice that in Lemma 2 we only give sufficient conditions for a mixed estimator to be efficient. Sadowsky and Bucklew (1990) formulated necessary and sufficient conditions for logarithmic efficiency of a mixed estimator of type (9), but these are rather restrictive, and many interesting problems such as the two-tailed problem (6) do not satisfy these sufficient conditions.

2.3 Expectations of functionals of Markov chains

Suppose that $(S_0 = 0, S_1, S_2, \ldots)$ is a Markov chain with jumps $X_1, X_2, \ldots$, namely $S_{k+1} = S_k + X_{k+1}$. Let the jump densities be $f_{k+1}(x_{k+1}|s_k)$. Thus the joint density of a sample path of jumps $(x_1, \ldots, x_n)$ is (using the Markov property)

$$f(x_1, \ldots, x_n) = \prod_{k=0}^{n-1} f_{k+1}(x_{k+1}|s_k).$$

Now let h be a function of sample paths of the form

$$h(x_1, \ldots, x_n) = \prod_{k=0}^{n-1} h_{k+1}(s_k, x_{k+1}),$$

for some functions $h_{k+1}(\cdot, \cdot)$ which depend on s_k and x_{k+1}, and suppose that we wish to determine $E[h(X_1, \ldots, X_n)]$. The following recursion is seen easily. Define random variables $Z_1, Z_2, \ldots, Z_n$ backwards by

$$\begin{aligned} Z_n &= 1 \quad \text{(w.p. 1)};\\ Z_k &= E[h_{k+1}(S_k, X_{k+1})Z_{k+1}|S_k] \quad \text{for } k = n-1, n-2, \ldots, 1. \end{aligned}$$

Then

$$E[h(X_1, \ldots, X_n)] = E[h_1(X_1)Z_1].$$

3 A sequential minimum cross-entropy scheme for tail probabilities

In this section we consider the one-tailed rare-event problem (5) for sums of i.i.d. random variables. Define the random walk (S_k) with jumps (X_k) by

$$S_0 = 0 \quad \text{and} \quad S_k = S_{k-1} + X_k = \sum_{j=1}^{k} X_j \quad (k = 1, 2, \ldots).$$

The random walk is a time-homogeneous Markov chain with state transitions (the jumps) that have probability density function $f(x)$ independent of the current state. We assume that $f(x)$ has light positive and negative tails, which means that $\int_{-\infty}^{\infty} e^{\theta x} f(x)\,dx < \infty$ for all θ in an open interval $(-\epsilon, \epsilon)$ containing zero.

We construct an importance sampling probability measure under which (S_k) becomes a time-inhomogeneous Markov chain with jumps whose distributions are state-dependent. We denote by $g_{k+1}(x|s)$ the conditional probability density function of the $k+1$-th jump X_{k+1}

given that $S_k = s$. Then the importance sampling density for the jump vector $\boldsymbol{X}(n)$ is clearly the product of these conditional time- and state-dependent densities, namely

$$g(x_1, \ldots, x_n) = \prod_{k=0}^{n-1} g_{k+1}(x_{k+1}|x_1, \ldots, x_k) = \prod_{k=0}^{n-1} g_{k+1}(x_{k+1}|s_k), \tag{10}$$

where $s_k = x_1 + \cdots + x_k$. We propose to construct the conditional density $g_{k+1}(x|s)$ via an MCE program. In fact, we formulate an MCE program for finding a conditional density $g_{k+1\to n}(x_{k+1}, \ldots, x_n|s)$ of all the 'future' jumps $\boldsymbol{X}(k+1, n) = (X_{k+1}, \ldots, X_n)$ given $S_k = s$. However, we only sample the $k+1$-th jump, giving the marginal

$$g_{k+1}(x_{k+1}|s) = \int g_{k+1\to n}(x_{k+1}, x_{k+2}, \ldots, x_n|s)\, dx_{k+2} \cdots dx_n.$$

This sequence of MCE programs is formed by repeatedly updating an original program (formulated prior to simulation) with the simulation history up to the current time.

Recalling the estimation target $P(S_n \geq bn)$, a natural constraint for the solution g is $E_g[S_n] \geq bn$. This has been argued to some extent in Sect. 14.2 of Bucklew (2004), where it is shown that the optimal rate of hitting the rare event should be around 0.5. Suppose that under g the random walk $(S_k)_k$ has i.i.d. jumps, then for sufficiently large n the state $S_n \overset{\mathrm{d}}{\approx} N(n\mu_g, n\sigma_g^2)$. Hence, when g is such that $E_g[S_n] = n\mu_g = bn$, then indeed $P_g(S_n \geq bn) \approx 0.5$. Although we allow a larger mean by the constraint $E_g[S_n] \geq bn$, and thus a larger hit rate, our solution will be exactly bn. Another objection against shifting the mean jump under g to far is the phenomenon of underestimation (Smith 2001). If we temporarily drop all subscripts of densities in the associated MCE program, after k steps it becomes

$$\begin{aligned} &\inf_{g\geq 0} \mathcal{D}_{\mathrm{KL}}(g|f) \\ &\text{s.t.} \quad \int g(\boldsymbol{x})\, d\boldsymbol{x} = 1, \\ &\qquad E_g\left[\sum_{j=k+1}^{n} X_j\right] \geq bn - s. \end{aligned} \tag{11}$$

The solution to this program is

$$g(x_{k+1}, \ldots, x_n|s) = f(x_{k+1}, \ldots, x_n) \exp\left(\lambda_0 + \lambda_1 \sum_{j=k+1}^{n} x_j\right),$$

where λ_0, λ_1 solve the corresponding dual program

$$\sup_{\lambda_0,\lambda_1} \lambda_0 + (bn - s)\lambda_1 - e^{\lambda_0} E_f\left[\exp\left(\lambda_1 \sum_{j=k+1}^{n} X_j\right)\right],$$

subject to $\lambda_1 \geq 0$. For working out the solution, we let $\mu = E_f[X]$ be the mean jump, and $\psi(\theta) = \log E_f[\exp(\theta X)]$ be the cumulant generating function of a single jump (under the original probability density $f(x)$), to get

$$g_{k+1}(x|s) = f(x)e^{\lambda_1 x - \psi(\lambda_1)}, \tag{12}$$

where λ_1 satisfies (note $k = 0, 1, \ldots, n-1$)

$$\begin{cases} \psi'(\lambda_1) = \frac{bn-s}{n-k}, & \text{if } \frac{bn-s}{n-k} \geq \mu, \\ \lambda_1 = 0, & \text{otherwise.} \end{cases} \tag{13}$$

Notice that in the first case the conditional density of the jump X_{k+1} is an exponentially tilted version of its original density $f(x)$ such that its mean becomes the average jump size to reach the rare event, and that in the latter case the conditional density is the original density f. In other words, if the first k jumps happened to be so much larger than usual that the remaining jumps would reach the rare event on average when operated under f, the next jump is indeed generated by f. In that case we say that the tilting is turned off in the importance sampling scheme.

We will write $\lambda_1 = \lambda_1(k, s)$ to explicitly express the dependence of the change of measure on time and state. From (13) we see that

$$\lambda_1(k, s) = (\psi')^{-1}\left(\frac{bn - s}{n - k} \vee \mu\right), \tag{14}$$

which again reflects 'turning off tilting' when appropriate.

Remark 3 We would get the same solution without the ability to 'turn off tilting', i.e., $\lambda_1(k, s) = (\psi')^{-1}((bn - s)/(n - k))$ for all times and states, if we had considered the MCE program (11) with the inequality symbol replaced by one for equality. This can be seen easily by following the steps of the construction of the algorithm.

3.1 Logarithmic efficiency

The importance sampling estimator associated with the sequential MCE approach is

$$\begin{aligned} Y(n) &= L(\boldsymbol{X}(n)) 1\{\boldsymbol{X}(n) \in A(n)\} \\ &= \left(\prod_{k=0}^{n-1} \frac{f(X_{k+1})}{g_{k+1}(X_{k+1}|S_k)}\right) 1\{S_n \geq bn\} \\ &= \left(\prod_{k=0}^{n-1} \exp\Big(-\lambda_1(k, S_k)X_{k+1} + \psi(\lambda_1(k, S_k))\Big)\right) 1\{S_n \geq bn\} \\ &= \exp\left(-\sum_{k=0}^{n-1}\Big(\lambda_1(k, S_k)X_{k+1} - \psi(\lambda_1(k, S_k))\Big)\right) 1\{S_n \geq bn\}. \end{aligned}$$

In Theorem 1 we shall prove that this estimator is logarithmically efficient. For that purpose, we note that, by Cramér's theorem (Dembo and Zeitouni 1998, Sect. 2.2), the tail probabilities $P(A(n)) = P(S_n \geq bn) = P(S_n/n \geq b)$ satisfy the large deviations limit

$$\lim_{n\to\infty} \frac{1}{n} \log P(A(n)) = -I(b),$$

where $I(b) = \sup_\theta(b\theta - \psi(\theta))$, and $b > \mu = E[X]$.

Consider the Markov chain $(S_k)_{k=0}^n$ when its jumps (X_k) have the importance sampling densities $g_k(x|s)$. We scale both time and space by n, and get a continuous process $\{s_n(t) : 0 \le t \le 1\}$ by linear interpolation, i.e., $s_n(t) = S_k/n$ if $t = k/n$. When $n \to \infty$ we obtain its (deterministic) fluid limit $\{y(t) : 0 \le t \le 1\}$, i.e., for any $\epsilon > 0$

$$\lim_{n\to\infty} P\Big(\sup_{0\le t\le 1} |s_n(t) - y(t)| < \epsilon\Big) = 1.$$

This limit holds because the jump densities $g_k(x|s)$ are Lipschitz continuous in the state s, which follows from (12) and (14), see Ethier and Kurtz (1986, Sect. 11.2). The fluid limit satisfies the ODE

$$y'(t) = \frac{b - y(t)}{1 - t}, \qquad y(0) = 0.$$

The solution is easily seen to be $y(t) = bt$. When we would determine the fluid limit associated to the importance sampling algorithm with constant tilting (3) which we described in the Introduction, we would obtain the same limit. This may be explained by noticing that the importance sampling densities in the two algorithms are constructed by a similar minimum cross-entropy program, albeit static versus sequentially adaptive.

Theorem 1 *The importance sampling estimator $Y(n)$ is logarithmically efficient.*

Proof As a consequence of the fluid limit, for any $0 < t < 1$ and $k = [tn]$, we have the approximation $S_k = bk + o_P(n)$ as $n \to \infty$, which means that

$$\lim_{n\to\infty} P\left(\left|\frac{S_k - bk}{n}\right| < \epsilon\right) = 1. \tag{15}$$

Moreover, the rate of convergence is exponential, thus, particularly

$$\lim_{C\to\infty} \limsup_{n\to\infty} \frac{1}{n} \log P\left(\frac{S_k - bk}{n} \le -C\right) = -\infty. \tag{16}$$

Now consider the following modification of the importance sampling scheme by choosing the tilting factor $\lambda_1(k, s)$ (denoted by $\lambda_1^{\mathrm{mod}}(k, s)$) according to

$$\begin{cases} \lambda_1^{\mathrm{mod}}(k,s) = (\psi')^{-1}(\frac{bn-s}{n-k}), & \text{if } \mu \le \frac{bn-s}{n-k} \le m, \\ \lambda_1^{\mathrm{mod}}(k,s) = 0, & \text{if } \frac{bn-s}{n-k} \le \mu, \\ \lambda_1^{\mathrm{mod}}(k,s) = (\psi')^{-1}(m), & \text{if } \frac{bn-s}{n-k} \ge m. \end{cases}$$

The new parameter m is larger than b, and later on we shall specify it. Because of (15), $\lambda_1^{\mathrm{mod}}(k, S_k) = (\psi')^{-1}(b) + o_P(1)$, where the error term $o_P(1)$ is uniformly bounded (in k and n) almost surely by some constant. Similarly, $\psi(\lambda_1(k, S_k)) = \psi((\psi')^{-1}(b)) + o_P(1)$ with a uniformly bounded (by a constant) error term. The constant bounds depend on the new parameter m. Now, since $(\psi')^{-1}(b)$ is positive we can obtain an upper bound on the second moment of the estimator $Y^{\mathrm{mod}}(n)$ as follows:

$$E[(Y^{\mathrm{mod}}(n))^2]$$
$$= E\left[\exp\left(-2\sum_{k=0}^{n-1}\left(\lambda_1^{\mathrm{mod}}(k, S_k)X_{k+1} - \psi(\lambda_1^{\mathrm{mod}}(k, S_k))\right)\right) 1\{S_n \ge bn\}\right]$$

$$
\begin{aligned}
&= E\left[\exp\left(-2\Big((\psi')^{-1}(b)S_n - n\psi((\psi')^{-1}(b)) + o_P(n)\Big)\right)1\{S_n \geq bn\}\right] \\
&\leq E\left[\exp\left(-2n\Big((\psi')^{-1}(b)b - \psi((\psi')^{-1}(b))\Big) + o_P(n)\right)1\{S_n \geq bn\}\right] \\
&= E[\exp(-2nI(b) + o_P(n))1\{S_n \geq bn\}].
\end{aligned}
$$

The resulting error term $o_P(n)$ satisfies

$$
\left|\frac{o_P(n)}{n}\right| \leq \text{constant (a.s.)} \quad \text{and} \quad \frac{o_P(n)}{n} \xrightarrow{P} 0 \quad (\text{if } n \to \infty).
$$

Thus we obtain

$$
\limsup_{n\to\infty} \frac{1}{n} \log E[(Y^{\text{mod}}(n))^2] \leq -2I(b). \tag{17}
$$

Let us return to our original importance scheme, and let us denote

$$
Z_n = \sum_{k=0}^{n-1} \Big(\lambda_1(k, S_k)X_{k+1} - \psi(\lambda_1(k, S_k))\Big).
$$

Consider the following conditioning of the second moment of the estimator:

$$
\begin{aligned}
E[(Y(n))^2] &= E\left[e^{-2Z_n}1\{S_n \geq bn\}\right] \\
&= E\left[e^{-2Z_n}1\{S_n \geq bn\}1\left\{\max_{k=0,\ldots,n-1} \frac{bn - S_k}{n-k} \leq m\right\}\right] \\
&\quad + E\left[e^{-2Z_n}1\{S_n \geq bn\}1\left\{\max_{k=0,\ldots,n-1} \frac{bn - S_k}{n-k} > m\right\}\right].
\end{aligned}
$$

The tilting parameter λ_1 is bounded below by 0 in this original change of measure, and thus the first term is equal to the second moment in the modified scheme $E[(Y^{\text{mod}}(n))^2]$. For the second term we notice that (after a few manipulations)

$$
\frac{bn - s}{n-k} > m \quad \Longleftrightarrow \quad \frac{s - bk}{n} \leq (b-m)\left(1 - \frac{k}{n}\right) \to -\infty \quad \text{if } m \to \infty.
$$

Using (16) we obtain also

$$
\lim_{m\to\infty} \limsup_{n\to\infty} \frac{1}{n} \log E\left[e^{-2Z_n}1\{S_n \geq bn\}1\left\{\max_{k=0,\ldots,n-1} \frac{bn - S_k}{n-k} > m\right\}\right] = -\infty,
$$

thus we can choose m so large that

$$
\limsup_{n\to\infty} \frac{1}{n} \log E\left[e^{-2Z_n}1\{S_n \geq bn\}1\left\{\max_{k=0,\ldots,n-1} \frac{bn - S_k}{n-k} > m\right\}\right] \leq -2I(b),
$$

to conclude

$$
\begin{aligned}
&\limsup_{n\to\infty} \frac{1}{n} \log E[(Y(n))^2] \\
&\quad \leq \limsup_{n\to\infty} \frac{1}{n} \log E[(Y^{\text{mod}}(n))^2] \\
&\qquad \vee \limsup_{n\to\infty} \frac{1}{n} \log E\left[e^{-2Z_n}1\{S_n \geq bn\}1\left\{\max_{k=0,\ldots,n-1} \frac{bn - S_k}{n-k} > m\right\}\right] = -2I(b).
\end{aligned}
$$

Using Jensen's inequality, we obtain the same lower bound for the liminf. Thus we have established the large deviations limit for the second moment, and as before we get efficiency by noting that

$$\lim_{n\to\infty} \frac{\log E[(Y(n))^2]}{\log E[Y(n)]} = \lim_{n\to\infty} \frac{\frac{1}{n}\log E[(Y(n))^2]}{\frac{1}{n}\log E[Y(n)]}$$
$$= \frac{\lim_{n\to\infty}\frac{1}{n}\log E[(Y(n))^2]}{\lim_{n\to\infty}\frac{1}{n}\log E[Y(n)]} = \frac{-2I(b)}{-I(b)} = 2. \qquad \square$$

Remark 4 Clearly the same fluid limit applies to the process that we would get by not turning off the tilting, see Remark 3. Thus, the associated estimator in that case is also logarithmically efficient. This seems to be an important point. The proof of Theorem 1 suggests that any importance sampling scheme for which the fluid scaling admits a fluid limit $y(t) = bt$, is logarithmically efficient. The limit path is a straight line to the rare event and stays away from the path $\phi(t) = (b-\mu)+\mu t$ which is a straight line to the rare event starting high enough at $\phi(0) = b - \mu > 0$, and running at the natural drift μ. In our algorithm, when the scaled simulated process deviates far from the limit path and hits $\phi(t)$, the tilting is turned off. Clearly, the larger the rarity parameter n becomes, the more the scaled simulated process follows the limit path, and thus, the less likely we have to turn off the tilting.

Consequently, considering the criterion the ratio $\log E[(Y(n))^2]/\log E[Y(n)]$ approaching 2 as $n \to \infty$ (see (4)), all importance sampling schemes with fluid limit $y(t) = bt$ are equivalent. Then the question arises what the point is to design other importance sampling schemes. The two main reasons refer to the other performance measures of estimators: relative error, and the speed (computer time) of the algorithm. In our examples we shall see that there might be huge differences in these criteria for logarithmically efficient estimators of the same quantity.

3.2 Strong efficiency with Gaussian jumps

In this section we assume that the jumps (X_k) are $N(0, 1)$ distributed. We shall show that the importance sampling estimator associated with the sequential MCE algorithm has bounded relative error, provided we give the last jump X_n the original density $f(x)$ conditioned that $X_n \geq bn - S_{n-1}$, which makes the rare event certain to occur.

In order to show bounded relative error, we rely on the results of Blanchet and Glynn (2006) who developed a state- and time-dependent importance sampling algorithm with bounded relative error for the same tail probability problem. In the next section, we shall give more details of their algorithm. For now, it suffices to mention that their state process (S_k) becomes a time-inhomogeneous Markov chain with jumps $X_{k+1} = S_{k+1} - S_k$, which have a normal distribution with mean $(bn - s)/(n - k)$ and variance $(n - k - 1)/(n - k)$, given current state $S_k = s$. This is the case for the first $n - 1$ jumps. The last jump X_n is realised from the original density $f(x)$ conditioned that $X_n \geq bn - S_{n-1}$. If we denote the resulting joint density of the jumps by $\hat{g}(\boldsymbol{x})$, and the associated likelihood ratio by $\hat{L}(\boldsymbol{x}) = f(\boldsymbol{x})/\hat{g}(\boldsymbol{x})$, then for any realisation of the jumps,

$$\hat{L}(\boldsymbol{x})1\{s_n \geq nb\} \leq c\frac{1}{\sqrt{n}}\exp(-nI(b)), \tag{18}$$

where c is some finite constant independent of n (Blanchet and Glynn 2006).

Theorem 2 *Assume that the jumps (X_k) are standard Gaussian. Then the importance sampling estimator associated with the sequential MCE scheme modified to have the conditional last jump has bounded relative error.*

Proof Firstly, in addition to the change to the last jump, we consider an adapted MCE importance sampling scheme in which tilting cannot be turned off (see Remark 3). The resulting joint density of the jumps is denoted by $g^{\mathrm{ad}}(\boldsymbol{x})$, and the associated likelihood ratio by $L^{\mathrm{ad}}(\boldsymbol{x}) = f(\boldsymbol{x})/g^{\mathrm{ad}}(\boldsymbol{x})$. Furthermore let

$$\mu_k(s) = \frac{bn - s}{n - k}.$$

Using the product property (10) of the joint densities $g^{\mathrm{ad}}(\boldsymbol{x})$ and $\hat{g}(\boldsymbol{x})$ we obtain:

$$\begin{aligned}
\frac{L^{\mathrm{ad}}(\boldsymbol{x})}{\hat{L}(\boldsymbol{x})} &= \prod_{k=0}^{n-2} \frac{\hat{g}_{k+1}(x_{k+1}|s_k)}{g^{\mathrm{ad}}_{k+1}(x_{k+1}|s_k)} \\
&= \prod_{k=0}^{n-2} \frac{\sqrt{n-k}}{\sqrt{n-k-1}} \exp\left(-\frac{1}{2}\frac{n-k}{n-k-1}(x_{k+1} - \mu_k(s_k))^2 + \frac{1}{2}(x_{k+1} - \mu_k(s_k))^2 \right) \\
&= \left(\prod_{k=0}^{n-2} \frac{\sqrt{n-k}}{\sqrt{n-k-1}} \right) \prod_{k=0}^{n-2} \exp\left(-\frac{1}{2}\frac{1}{n-k-1}(x_{k+1} - \mu_k(s_k))^2 \right).
\end{aligned}$$

The first factor works out to $\prod_{k=0}^{n-2} \sqrt{n-k}/\sqrt{n-k-1} = \sqrt{n}$. Hence, by squaring the ratio and taking the expectation w.r.t. the importance sampling density g^{ad}, we get

$$\frac{1}{n} E_{g^{\mathrm{ad}}}\left[\left(\frac{L^{\mathrm{ad}}(\boldsymbol{x})}{\hat{L}(\boldsymbol{x})} \right)^2 \right] = E_{g^{\mathrm{ad}}}\left[\prod_{k=0}^{n-2} \exp\left(-\frac{1}{n-k-1}(X_{k+1} - \mu_k(S_k))^2 \right) \right]. \tag{19}$$

This is of the form described in Sect. 2.3. When we work out the recursion given there, we first notice that it is easily verified by calculus that, when X is a $N(\mu, \sigma^2)$ random variable,

$$E\left[\exp(\theta(X - \mu)^2) \right] = \frac{1}{\sqrt{1 - 2\theta\sigma^2}},$$

for $\theta < 1/(2\sigma^2)$. When we apply this we get by induction to k, for $k = n-2, n-3, \ldots, 0$ (see Sect. 2.3)

$$\begin{aligned}
Z_k &= E_{g^{\mathrm{ad}}}\left[\exp\left(-\frac{1}{n-k-1}(X_{k+1} - \mu_k(S_k))^2 \right) Z_{k+1} \middle| S_k \right] \\
&= \prod_{j=k}^{n-2} \frac{1}{\sqrt{1 + 2/(n-j-1)}} \quad \text{(w.p. 1)},
\end{aligned}$$

because under the importance sampling density g^{ad}, jump X_{k+1} given state S_k is $N(\mu_k(S_k), 1)$. Thus, the product (19) becomes

$$\prod_{k=0}^{n-2} \frac{1}{\sqrt{1 + 2/(n-k-1)}} = \prod_{k=0}^{n-2} \frac{\sqrt{n-k-1}}{\sqrt{n-k+1}} = \frac{\sqrt{2}}{\sqrt{n(n+1)}},$$

yielding

$$E_{g^{\mathrm{ad}}}\left[\left(\frac{L^{\mathrm{ad}}(\boldsymbol{X})}{\hat{L}(\boldsymbol{X})}\right)^2\right] = \frac{n\sqrt{2}}{\sqrt{n(n+1)}} \le \sqrt{2}.$$

We combine this with the bounding of the ratio $\hat{L}$ in (18) to obtain

$$\begin{aligned} E_{g^{\mathrm{ad}}}\left[\left(L^{\mathrm{ad}}(\boldsymbol{X})\right)^2 1\{S_n \ge nb\}\right] &= E_{g^{\mathrm{ad}}}\left[\left(\frac{L^{\mathrm{ad}}(\boldsymbol{X})}{\hat{L}(\boldsymbol{X})}\right)^2 \left(\hat{L}(\boldsymbol{X})\right)^2 1\{S_n \ge nb\}\right] \\ &\le \sqrt{2}c^2 \frac{1}{n}\exp(-2nI(b)). \end{aligned}$$

Finally, we see that the adapted importance sampling scheme has bounded relative error by observing that (Blanchet and Glynn 2006)

$$\limsup_{n\to\infty} \frac{\exp(-2nI(b))/n}{P(S_n \ge nb)} < \infty.$$

Now let us return to our original importance sampling scheme with tilting that can be turned off (but still with the conditional last jump). The corresponding importance sampling density is $g(\boldsymbol{x})$ and its associated likelihood ratio is $L(\boldsymbol{x}) = f(\boldsymbol{x})/g(\boldsymbol{x})$. Clearly it suffices to show that

$$E_f\left[L(\boldsymbol{X})1\{S_n \ge nb\}\right] \le K E_f\left[L^{\mathrm{ad}}(\boldsymbol{X})1\{S_n \ge nb\}\right],$$

for some finite constant K, where we take expectations with respect to the original density $f(\boldsymbol{x})$. These likelihood ratios are

$$\begin{aligned} L(\boldsymbol{X}) &= \left(\prod_{k=0}^{n-2} \frac{f(X_{k+1})}{g_{k+1}(X_{k+1}|S_k)}\right) \bar{F}(bn - S_{n-1}), \\ L^{\mathrm{ad}}(\boldsymbol{X}) &= \left(\prod_{k=0}^{n-2} \frac{f(X_{k+1})}{g^{\mathrm{ad}}_{k+1}(X_{k+1}|S_k)}\right) \bar{F}(bn - S_{n-1}), \end{aligned}$$

wherein we have substituted the likelihood ratio of the conditional last jump (and used the notation $\bar{F}(x) = P(X > x)$). The expectations

$$E_f\left[L(\boldsymbol{X})1\{S_n \ge nb\}\right] \quad \text{and} \quad E_f\left[L^{\mathrm{ad}}(\boldsymbol{X})1\{S_n \ge nb\}\right],$$

are calculated recursively as in Sect. 2.3:

$$E_f\left[L(\boldsymbol{X})1\{S_n \ge nb\}\right] = E_f\left[\prod_{k=0}^{n-1} h_{k+1}(S_k, X_{k+1})\right],$$

with

$$h_{k+1}(s_k, x_{k+1}) = \frac{f(x_{k+1})}{g_{k+1}(x_{k+1}|s_k)}, \quad k = 0, \ldots, n-2,$$

and

$$h_n(s_{n-1}, x_n) = \bar{F}(bn - s_{n-1})1\{s_{n-1} + x_n \ge nb\}.$$

The recursion gives random variables Z_k for the original importance sampling, and Z_k^{ad} for the adapted version with forced tilting. The backwards recursion starts with $Z_n = Z_n^{\mathrm{ad}} = 1$, and $Z_{n-1} = Z_{n-1}^{\mathrm{ad}}$ with probability 1. Assume $Z_k \le Z_k^{\mathrm{ad}}$ with probability 1. Then one can show that

$$\begin{aligned} Z_{k-1} &= E_f[h_k(S_{k-1}, X_k) Z_k | S_{k-1}] = E_f\left[\frac{f(X_k)}{g_k(X_k|S_{k-1})} Z_k \Big| S_{k-1}\right] \\ &\le E_f\left[\frac{f(X_k)}{g_k(X_k|S_{k-1})} Z_k^{\mathrm{ad}} \Big| S_{k-1}\right] \\ &\overset{\text{(i)}}{\le} E_f\left[\frac{f(X_k)}{g_k^{\mathrm{ad}}(X_k|S_{k-1})} Z_k^{\mathrm{ad}} \Big| S_{k-1}\right] = Z_{k-1}^{\mathrm{ad}}. \end{aligned}$$

The inequality (i) follows from the following reasoning. Whenever $\mu_k(s) > 0$ (where $k = 0, \ldots, n-2$), the jump densities are the same, so $g_{k+1}(x|s) = g_{k+1}^{\mathrm{ad}}(x|s)$ in this case. However, when $\mu_k(s) \le 0$, the $k+1$-th factor contributes just 1 to the product in L because $g_{k+1}(x|s) = f(x)$. For the adapted version, the jump has an exponentially tilted density given in (12) with $\psi(\theta) = \frac{1}{2}\theta^2$ in the standard Gaussian case, giving $\psi'(\theta) = \theta$. Hence, the contribution to the product in L^{ad} is

$$\begin{aligned} &E_f\left[\frac{f(X_{k+1})}{g_{k+1}^{\mathrm{ad}}(X_{k+1}|S_k)} \Big| S_k = s\right] \\ &\quad = E_f\left[\exp\Big(-\mu_k(s) X_{k+1} + \psi(\mu_k(s))\Big)\right] \\ &\quad = \exp\big(\psi(-\mu_k(s)) + \psi(\mu_k(s))\big) = \exp(\mu_k^2(s)) \ge 1. \end{aligned}$$

□

Remark 5 In the case of general Gaussian jumps, i.e., those that are $N(\mu, \sigma^2)$ distributed, we again obtain bounded relative error of the algorithm. This can be seen by using the distributional relationship $N(\mu, \sigma^2) \overset{\mathrm{d}}{=} \mu + \sigma N(0, 1)$ and then following the line of reasoning above.

3.3 Simulation experiments

We have experimented with the core MCE algorithm given above, its adaptations, and with efficient algorithms from literature. A brief outline of each is given below.

SEQ-MCE-IN. Our core algorithm with state- and time-dependent exponential tilting based on the MCE program (11). This algorithm has the property that tilting is turned off when unnecessary, but does not use the conditional last jump. We proved logarithmic efficiency for this algorithm in Sect. 3.1.

SEQ-MCE-IN-COND. Similar to the core algorithm, but with the conditional last jump. In the case of Gaussian jumps this algorithm is proven to have bounded relative error (Sect. 3.2).

SEQ-MCE-EQ. This algorithm again implements the same state- and time-dependent exponential tilting, but without the ability to turn it off. It is based on the MCE program (11) with equality constraints and is logarithmically efficient, see Remark 3 and Remark 4. (Note that if the last jump is made conditional, then this algorithm is also proven to have bounded relative error for Gaussian jumps in Sect. 3.2.)

STATIC. The classical state-independent algorithm using exponential tilting with the optimal tilting parameter $(\psi')^{-1}(b)$. It is well known that this algorithm is logarithmically efficient (Bucklew 2004).

BG-EQ-COND. A state- and time-dependent algorithm for Markov chains given in Blanchet and Glynn (2006); L'Ecuyer et al. (2008). We give an outline of the algorithm because we found a slightly different implementation of it. Given current state $S_j = s$, the next jump X_{j+1} is realised from a distribution of the form

$$P(X_{j+1} \in (x, x + dx) | S_j = s) = \frac{f(x) v_{j+1}(s + x)}{w_j(s)} dx, \tag{20}$$

where $w_j(s)$ is the normalising constant, and where $v_{j+1}(y)$ is an approximation of $P(S_n \geq nb | S_{j+1} = y) = P(\sum_{i=j+2}^{n} X_i \geq nb - y)$. In case the (original) jumps have a $N(\mu, \sigma^2)$ distribution, Blanchet and Glynn (2006) suggest using

$$v_j(y) = \frac{\exp(-(n-j) I((bn - y)/(n - j)))}{\sqrt{n - j}},$$

where $I(\cdot)$ is the Legendre-Fenchel transform of a jump X, i.e., $I(x) = \sup_\theta(\theta x - \psi(\theta))$. Using this, the right-hand side of (20) works out to be a normal density with mean $(bn - s)/(n - j)$ and variance $\sigma^2(n - j - 1)/(n - j)$. (This is where our calculations differ from Blanchet and Glynn (2006) who found a normal density with mean $(bn - s)/(n - j - 1)$ and variance $\sigma^2(n - j)/(n - j - 1)$.) This is done for the first $n - 1$ jumps. The last jump X_n is realised from the original density $f(x)$ conditioned that $X_n \geq bn - S_{n-1}$. Notice that, unlike the SEQ-MCE algorithms, both the mean and variance are modified under g. As was previously mentioned, this scheme was shown to give bounded relative error for Gaussian jumps (Blanchet and Glynn 2006).

BG-IN-COND. The same as the BG-EQ-COND algorithm, but turning off tilting when appropriate.

We have applied these algorithms for i.i.d. jumps with the following distributions.

- Bernoulli (p) on $\{-1, 1\}$, i.e., $P(X = 1) = p$, $P(X = -1) = 1 - p$.
- Laplace (κ), i.e., $f(x) = \frac{1}{2}\kappa e^{-\kappa|x|} x \in \mathbb{R}$.
- Normal (μ, σ^2).
- Double Coxian-2, i.e., $X = \Delta\xi_1 - (1 - \Delta)\xi_2$ with Δ Bernoulli (p) on $\{0, 1\}$, and ξ_1 and ξ_2 are independent Coxian-2 distributed random variables on $[0, \infty)$, and independent of Δ. The Coxian-2 density $f(x)$ is defined for $x \geq 0$, and has three parameters $b \in [0, 1]$, $\mu_1 > 0$, $\mu_2 > 0$:

$$f(x) = (1 - b)\mu_1 e^{-\mu_1 x} + b\mu_1 e^{-\mu_1 x} * \mu_2 e^{-\mu_2 x},$$

where $*$ means convolution of the two exponential densities. The Coxian-2 distribution is used to model higher variances, see Appendix B in Tijms (2003). In the same manner, one can generalize the Laplace distribution to become double Exponential.

As for parameters: the Normal standard, and the other three symmetric with $p = 0.5$. Furthermore, we took $\kappa = 1$ in the Laplace distribution, which would mean for the associated single Exponential ξ_1, expectation $E[\xi_1] = 1$ and squared coefficient variation $\text{Var}[\xi_1]/(E[\xi_1])^2 = 1$. In the double Coxian-2 we set the parameters such that the single Coxian-2 ξ_1 has expection $= 1$, and squared coefficient variation $= 5$.

The BG-EQ-COND and BG-IN-COND algorithms were implemented for the Normal case only, since it was not clear how to generalise the method to other distributions.

After each simulation experiment we collect three (estimated) performance measures of the importance sampling estimator $Y(n)[k]$ of $\ell(n)$ based on k samples:

- RHW: the relative half width of the 95% confidence interval for $Y(n)[k]$, namely

$$1.96\sqrt{\mathrm{Var}[Y(n)[k]]}/E[Y(n)[k]].$$

- RAT: the logarithmic efficiency ratio, cf. (4),

$$\log E[(Y(n)[k])^2]/\log E[Y(n)[k]].$$

- EFF: the efficiency which takes into account both the variance of the estimator and the total execution time (in seconds on a PC with a 2.4 GHz CPU running under Linux) of the simulation:

$$\frac{1}{\mathrm{Var}[Y(n)[k]] \times \mathrm{CPU}[Y(n)[k]]}.$$

Better performance is obtained by smaller RHW, higher RAT, and larger EFF.

3.4 Observations

We find that the algorithms which can turn off tilting perform better than their counterparts which cannot, that is, SEQ-MCE-IN vs. SEQ-MCE-EQ, and BG-IN-COND vs. BG-EQ-COND (Fig. 1). This is due to the fact that the likelihood ratio can become disproportionately large when the tilting parameter $\lambda_1 < 0$. For instance, in the case of a Laplacian jump X, straightforward calculus shows that the likelihood ratio of X equals

$$L(X) = \frac{f(X)}{g(X)} = M(\lambda_1)e^{-\lambda_1 X} = \frac{\kappa^2}{\kappa^2 - \lambda_1^2}e^{-\lambda_1 X},$$

which becomes large when $\lambda_1 < 0$ and the jump $X > 0$. Furthermore, not surprisingly, the algorithm SEQ-MCE-IN-COND with the conditional last jump performs better than its counterpart SEQ-MCE-IN (Figs. 2 and 5).

Empirically we found that, out of the two proven strongly efficient algorithms, our SEQ-MCE-IN-COND algorithm gives better performance than BG-IN-COND (Fig. 3). Although we could only prove logarithmic efficiency of our original SEQ-MCE-IN algorithm (without the conditional last jump), the simulation results seem to indicate bounded relative error as well (Figs. 2 and 5). Typical logarithmically efficient behaviour is observed in the performance of the STATIC algorithm, which has increasing RHW (Fig. 4). Summarising, the best performance is obtained by the SEQ-MCE-IN-COND algorithm, and the worst by the STATIC algorithm.

To illustrate, we first show results for the Normal case with $\mu = 0$, $\sigma^2 = 1$ (standard Normal jumps) and overflow level $b = 2/3$, sample size $k = 10000$, and n spanning the range 50–1000. The results presented are the averages of 100 repetitions of these simulations. In the following figures we plot the graphs of RHW and RAT only, and do not graph the efficiencies EFF because these increase exponentially with roughly identical rates for all algorithms (see Table 1). (Note however that these small differences can account for an appreciable reduction in CPU time for the same variance. For instance, when $n = 1000$ as

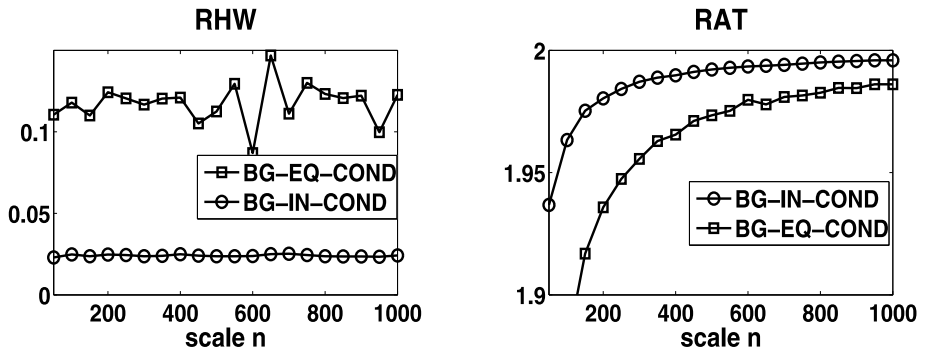

Fig. 1 Forced vs. optional tilting for the BG algorithm on the one-tailed problem with Gaussian jumps

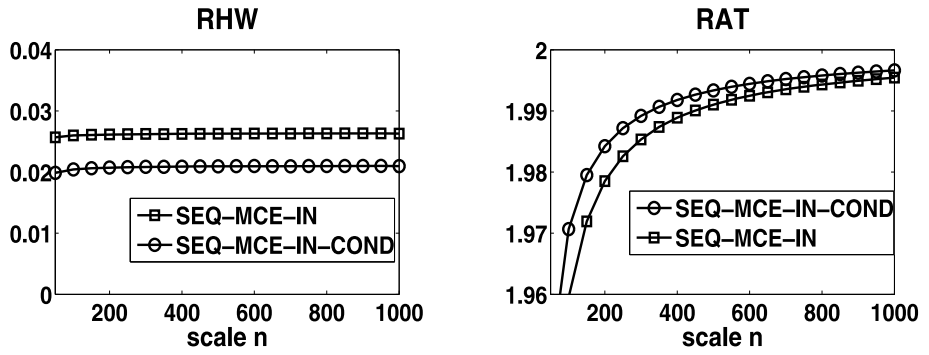

Fig. 2 The SEQ-MCE-IN algorithm with vs. without conditional last jump on the one-tailed problem with Gaussian jumps

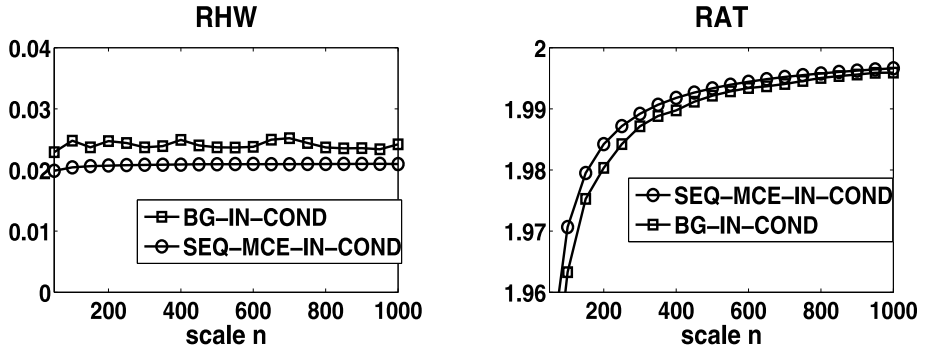

Fig. 3 Comparison of the best BG and SEQ-MCE algorithms on the one-tailed problem with Gaussian jumps

in Table 1, the slowest of our algorithm implementations takes more than 30 times as long as the fastest.)

Our experiments with other light-tailed jump distributions, such as the Laplacian, Bernoulli and double Coxian-2, as mentioned above, gave the same indication that the SEQ-

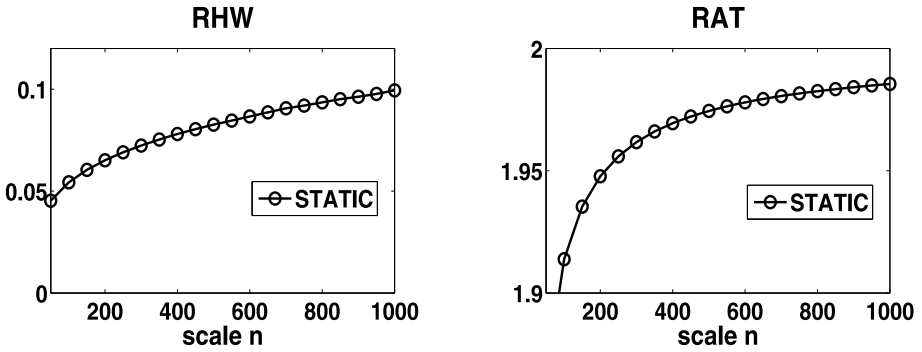

Fig. 4 Performance of the state-independent STATIC estimator for the one-tailed problem with Gaussian jumps

Table 1 The exponential growth rate of the performance EFF of all estimators for the one-tailed problem with Gaussian jumps at $n = 1000$, i.e. $\frac{1}{n} \log \mathrm{EFF}$

SEQ-MCE-IN	0.4610
SEQ-MCE-IN-COND	0.4615
SEQ-MCE-EQ	0.4583
STATIC	0.4586
BG-IN-COND	0.4605
BG-EQ-COND	0.4580

MCE-IN algorithms (with or without the last conditional jump) yield bounded relative error, see Fig. 5. Clearly, in case of the Bernoulli jump it is not always feasible to reach the rare event by conditioning the last jump. However (for the Bernoulli case), the SEQ-MCE-EQ algorithm generates realizations with a constant likelihood ratio, and thus its associated estimator has zero-variance.

As noted above, it was not clear how to implement the BG algorithms for the other jump distributions, and so we cannot comment on the relative performance of the BG and SEQ-MCE-IN algorithms using jump distributions other than Gaussian.

4 The two-tailed problem

In this section we consider the two-tailed rare-event problem (6), i.e.,

$$\ell_n = P(\{S_n/n \le a\} \cup \{S_n/n \ge b\}),$$

where $a < \mu = E[X] < b$. We assume that $I(a) > I(b)$ for the large deviations rate function $I(\cdot)$. A strongly or logarithmically efficient algorithm is obtained by applying a mixed importance sampling estimator as defined in Sect. 2.2 when the conditions Lemma 1, or Lemma 2 respectively, are fulfilled.

Hence, mixing the importance sampling densities of MCE-SEQ-COND or BG-IN-COND gives bounded relative error for Gaussian jumps. Moreover, since all algorithms of Sect. 3.3 are logarithmically efficient, their associated mixed estimators are logarithmically efficient, provided the remaining conditions (a) and (c) of Assumption 1 are fulfilled. Let $A_1(n) = \{S_n/n \le a\}$ and $A_2(n) = \{S_n/n \ge b\}$, then condition (a) is clearly satisfied by

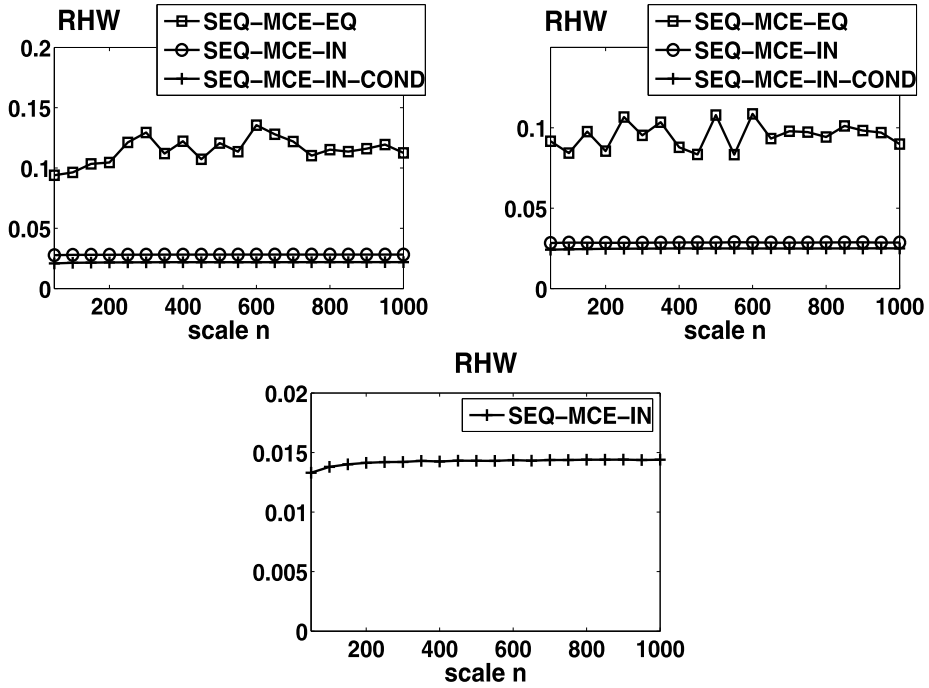

Fig. 5 The RHW performance of estimators for the one-tailed problem with Laplacian jumps (*top left*), double Coxian-2 jumps (*top right*) and Bernoulli jumps (*bottom*). Thresholds: $b = 1$ (Laplacian), $b = 1.5$ (Coxian), $b = 0.6$ (Bernoullian). Sample sizes: $k = 10000$ (Laplacian, and Coxian), $k = 5000$ (Bernoullian). Results are the averages of 100 experiments

application of Cramér's theorem for i.i.d. sums (Dembo and Zeitouni 1998, Sect. 2.2). Condition (c) holds for instance when the mixing probabilities p_j are constant (in n). Glasserman and Wang (1997) propose mixing probabilities dependent on the large deviations rate function (being the choice that asymptotically minimises the variance of the 'deterministic' estimator), namely

$$\begin{aligned} p_1(n) &= \frac{\exp(-nI(a) + o(n))}{\exp(-nI(a) + o(n)) + \exp(-nI(b) + o(n))} \\ &= \frac{1}{1 + \exp(n(I(a) - I(b)) + o(n))} \\ &\approx \exp(-n(I(a) - I(b))). \end{aligned} \tag{21}$$

However, these decay exponentially fast to zero. As a remedy one might introduce a cut-off threshold η and use

$$p_1(n) = \frac{1}{1 + \exp(n(I(a) - I(b)))} \vee \eta \tag{22}$$

in an implementation, for some small $0 < \eta < 1$.

4.1 Mixing probabilities using minimum cross-entropy

An alternate way to obtain mixing probabilities is to introduce the mixing random variable (see Sect. 2.2) by augmenting the original state space with $\Delta(n)$ so that it is independent of $\boldsymbol{X}(n)$, and solve an augmented MCE program. For the two-tailed problem, we define $\Delta(n)$ on $\{0, 1\}$, associating an outcome of 0 with the event $\{S_n/n \leq a\}$, and 1 with the event $\{S_n/n \geq b\}$. Writing $\pi(\delta) = P(\Delta(n) = \delta)$, the simultaneous probability density function of the mixing variable and all the jumps is given by

$$\tilde{f}(\delta, x_1, \ldots, x_n) = \pi(\delta) f(x_1, \ldots, x_n) = \pi(\delta) \prod_{j=1}^{n} f(x_j),$$

with $\delta \in \{0, 1\}$, and each $x_j \in \mathbb{R}$ as before.

To obtain a change of measure for $\Delta(n)$, we apply the machinery of Sect. 2.1 to solve the MCE program with 'mixture constraint'

$$E_{\tilde{g}}\left[(1 - \Delta(n))\,(a - S_n/n) + \Delta(n)\,(S_n/n - b)\right] \geq 0.$$

Notice that, given $\Delta(n) = \delta$, this constraint reduces to a constraint for one of the one-tailed component problems that form the two-tailed problem.

Solving the augmented MCE program proceeds as follows. First we rewrite the constraint function as

$$\begin{aligned} c(\delta, x_1, \ldots, x_n) &= (1 - \delta)\left(a - \frac{1}{n}\sum_{j=1}^{n} x_j\right) + \delta\left(\frac{1}{n}\sum_{j=1}^{n} x_j - b\right) \\ &= a(1 - \delta) - b\delta + \frac{1}{n}(2\delta - 1)\sum_{j=1}^{n} x_j. \end{aligned}$$

The solution to the MCE program is then

$$\begin{aligned} \tilde{g}(\delta, x_1, \ldots, x_n) &= \tilde{f}(\delta, x_1, \ldots, x_n) \exp\left(\lambda_0 + \lambda_1 c(\delta, x_1, \ldots, x_n)\right) \\ &= \pi(\delta)\left(\prod_{j=1}^{n} f(x_j)\right) \exp\left(\lambda_0 + \lambda_1 a(1 - \delta) - \lambda_1 b\delta + \frac{1}{n}\lambda_1(2\delta - 1)\sum_{j=1}^{n} x_j\right) \\ &= e^{\lambda_0}\left(\pi(\delta) \exp(\lambda_1 a(1 - \delta) - \lambda_1 b\delta)\right)\left(\prod_{j=1}^{n} f(x_j) \exp\left(\frac{1}{n}\lambda_1(2\delta - 1)x_j\right)\right). \end{aligned}$$

This is of the form $q(\delta)g(x_1, \ldots, x_n|\delta)$, which tells us that the probability density of the jumps under the importance sampling density depends on the outcome of the Bernoulli $\Delta(n)$. After some manipulation, we find that the biased Bernoulli probabilities are given by

$$\begin{aligned} p_1(n) &= q(0) = e^{\lambda_0}\pi(0)e^{\lambda_1 a}M(-\lambda_1/n)^n, \\ p_2(n) &= q(1) = e^{\lambda_0}\pi(1)e^{-\lambda_1 b}M(\lambda_1/n)^n, \end{aligned} \tag{23}$$

where $M(\theta) = E_f[\exp(\theta X)]$ is the moment generating function of a single jump under f. Again, this solution to the MCE program constitutes an exponential tilting by writing

$$p_1(n) \propto \exp(-n(\theta_n a - \psi(\theta_n))),$$

with tilting parameter $\theta_n = -\lambda_1/n$ dependent on the number of jumps. When $\psi'(\theta_n) = a$ we would get

$$p_1(n) \propto \exp(-nI(a)),$$

which coincides with the mixing probability given in (21). However, the exact numerical values are obtained by solving for the Lagrange multipliers λ_0, λ_1 in (23). It turns out that $\lambda_0 = 1/Q(\lambda_1)$, and λ_1 solves $Q'(\lambda_1) = 0$, where

$$Q(\lambda_1) = \big(\pi(0)e^{\lambda_1 a}M(-\lambda_1/n)^n + \pi(1)e^{-\lambda_1 b}M(\lambda_1/n)^n\big).$$

The equation $Q'(\lambda_1) = 0$ is equivalent to

$$\begin{aligned}&\pi(0)\exp\big(-n\big((-\lambda_1/n)a - \psi(-\lambda_1/n)\big)\big)\big(a - \psi'(-\lambda_1/n)\big)\\ &\quad - \pi(1)\exp\big(-n\big((\lambda_1/n)b - \psi(\lambda_1/n)\big)\big)\big(b - \psi'(\lambda_1/n)\big) = 0.\end{aligned}$$

Although this equation must be solved numerically, we see that indeed

$$p_1(n) \propto \exp\left(-nI(a) + o(n)\right) \quad \text{or} \quad p_2(n) \propto \exp\left(-nI(b) + o(n)\right),$$

as $n \to \infty$. Finally, the probabilities given in (23), or their asymptotic equivalents, can subsequently be used in any of the importance sample mixing schemes outlined in Sect. 2.2.

4.2 Importance sampling algorithms

We propose using a mixed importance sampling estimator (7) with mixing probabilities $p_j(n)$ given in the previous section in (23). However, given the outcome of the Bernoulli variable $\Delta(n)$ we apply the sequential MCE scheme of Sect. 3 to find the importance sampling densities of the jumps. Note that we only apply the SEQ-MCE-IN-COND algorithm to the two-tailed problem, as the performance of the other MCE-based algorithms was inferior on the one-tailed problems. The full specification of the MCE algorithm is given below.

Algorithm TWO-SEQ-MCE-IN-COND

1. *Generate* $\delta \in \{0, 1\}$ *from the density* (23).
2. *If* $\delta = 0$, *simulate the random walk* (S_k) *from* $S_0 = 0$ *up to* S_{n-1} *where jump* X_{k+1} $(k = 0, \ldots, n-2)$ *is generated from the tilted density* $g_{k+1}(x) = f(x)\exp(\theta x - \psi(\theta))$ *with tilting parameter* $\theta = (\psi')^{-1}((an - S_k)/(n-k) \wedge \mu)$, *and jump* X_n *is generated from the conditional distribution* $P_f(X \in \cdot | X \le an - S_{n-1})$.
3. *If* $\delta = 1$, *simulate the random walk* (S_k) *from* $S_0 = 0$ *up to* S_{n-1} *where jump* X_{k+1} $(k = 0, \ldots, n-2)$ *is generated from the tilted density* $g_{k+1}(x) = f(x)\exp(\theta x - \psi(\theta))$ *with tilting parameter* $\theta = (\psi')^{-1}((bn - S_k)/(n-k) \vee \mu)$, *and jump* X_n *is generated from the conditional distribution* $P_f(X \in \cdot | X \ge bn - S_{n-1})$.

At the beginning of this section we remarked that this is a logarithmically efficient importance sampling algorithm, provided that the mixing probabilities do not decay exponentially fast to zero. To ensure this we modify the algorithm slightly by cutting off the mixing probability $p_1(n)$ at η for an arbitrary $0 < \eta < 1$ as in (22). Moreover, in our implementation, we used the 'deterministic' equivalent of the algorithm, see Remark 1.

The importance sampling algorithm and its variation TWO-SEQ-MCE-EQ are applied to the three models mentioned in Sect. 3.3: Bernoulli, Laplace, and Normal distributed jumps.

We compare our algorithms with other logarithmically efficient algorithms for the two-tailed problem given in Glassermann and Wang (1997) and in Dupuis and Wang (2004, 2007), denoted TWO-STATIC, DW-SOL, and DW-SUBSOL, respectively. The two Dupuis-Wang methods have been developed specifically to deal with these more complex rare events, and thus we did not include them in our experiments of the one-tailed problem. We give a brief summary of the algorithms below.

TWO-SEQ-MCE-EQ. Similar to TWO-SEQ-MCE-IN-COND, without the ability to turn the tilting off, and without the conditional last jump.

TWO-STATIC. Mixed importance sampling estimator with state-independent exponentially tilted jump densities, where the tilting parameters are $(\psi')^{-1}(a)$ for the samples allocated to estimate $P(S_n \le an)$, and $(\psi')^{-1}(b)$ for the samples allocated to estimate $P(S_n \ge bn)$.

DW-SOL. This algorithm is based on the solution of an Isaacs equation (Dupuis and Wang 2004). The importance sampling algorithm is time- and state-dependent, in which jump X_{j+1} is realised from an exponentially tilted density

$$P(X_{j+1} \in (x, x+dx) | S_j = s) = f(x) \exp(\theta x - \psi(\theta))\, dx,$$

where the tilting parameter $\theta = \theta(s, j)$ depends on time j and state s as follows. Recall that the rare event is given by $A(n) = \{S_n/n \le a\} \cup \{S_n/n \ge b\}$ with $a < \mu < b$. Define for $x \in \mathbb{R}$ and $t \in [0, 1]$

$$U(x,t) = \inf_{\beta} \{(1-t) I(\beta) : x + (1-t)\beta \in A(n)\}.$$

Then the tilting parameters are

$$\theta(j,s) = -\frac{\partial}{\partial x} U(x,t)\Big|_{x=s/n, t=j/n}.$$

In words, this algorithm is doing the following. At any time it detects which of the two parts of the rare event is the most likely one, and then applies an exponential tilting of the next jump X_k in order to get there on average.

DW-SUBSOL. We give a short outline of the algorithm based on a subsolution of an Isaacs equation (Dupuis and Wang 2007). The importance sampling algorithm is time- and state-dependent in which each jump is realised from a mixture of exponentially tilted densities, i.e.,

$$P(X_{j+1} \in (x, x+dx) | S_j = s) = \sum_{i=1}^{2} p_i^{\delta} f(x) \exp(\theta_i x - \psi(\theta_i))\, dx.$$

The tilting parameters θ_i are fixed throughout the simulation, and the mixing probabilities p_i^{δ} depend on jump time $j+1$, state $S_j = s$, and so-called mollification parameter δ. We associate $i = 1$ with the event $\{S_n/n \le a = \beta_1\}$ and $i = 2$ with the event $\{S_n/n \ge b = \beta_2\}$. Finally, define subsolution/control pairs $(\bar{W}_i, \theta_i)$ for $i = 1, 2$ by

$$\bar{W}_i(x,t) = -2\theta_i x + 2\theta_i \beta_i - 2(1-t)\psi(\theta_i) \quad (x \in \mathbb{R}, t \in [0,1]),$$
$$\theta_i = (\psi')^{-1}(\beta_i).$$

Notice that these tilting parameters are the same as in the STATIC algorithm. Suppose that $S_j = s$, then set $x = s/n$ and $t = j/n$. The mixing probabilities to determine which tilting

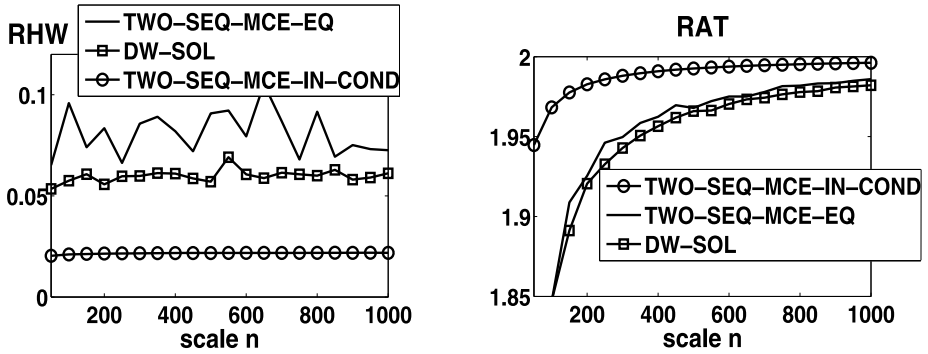

Fig. 6 Comparison of DW-SOL and SEQ-MCE algorithms (with forced and optional tilting) on the two-tailed problem with Gaussian jumps

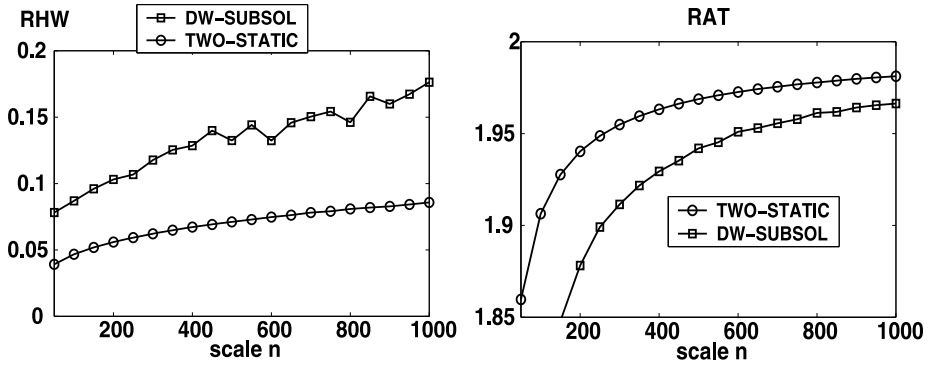

Fig. 7 Comparison of DW-SUBSOL and STATIC algorithms on the two-tailed problem with Gaussian jumps

will be used for jump X_{j+1} are

$$p_i^\delta = p_i^\delta(x,t) = \frac{\exp(-\bar{W}_i(x,t)/\delta)}{\sum_r \exp(-\bar{W}_r(x,t)/\delta)},$$

where $\delta > 0$.

We show the results for Gaussian distributed jumps with mean $\mu = 3$ and variance $\sigma^2 = 1$, and overflow levels $a = 2.4999$, $b = 3.5$ (Figs. 6 and 7; Table 2); and for Laplace distributed jumps, with parameter $\kappa = 1$, and with overflow levels $a = -1.25$, $b = 1$ (Figs. 8 and 9; Table 2). We used sample sizes of $k = 10000$ and let n span the range 50–1000. The experiments were performed with the TWO-SEQ-MCE-IN-COND algorithm using the mixing probabilities (23) cut off at $\eta = 0.01$, the DW-SOL algorithm, the DW-SUBSOL algorithm using mollification parameter $\delta = 0.001$, and the TWO-SEQ-MCE-EQ and TWO-STATIC algorithms. All experiments have been repeated 100 times from which we show the average performance.

We observe similar behaviour as before, however, notice particularly that the relative error of the TWO-SEQ-MCE-EQ is rather large compared to the others, although it seems to show bounded relative error. Also notice that DW-SOL seems to be strongly efficient

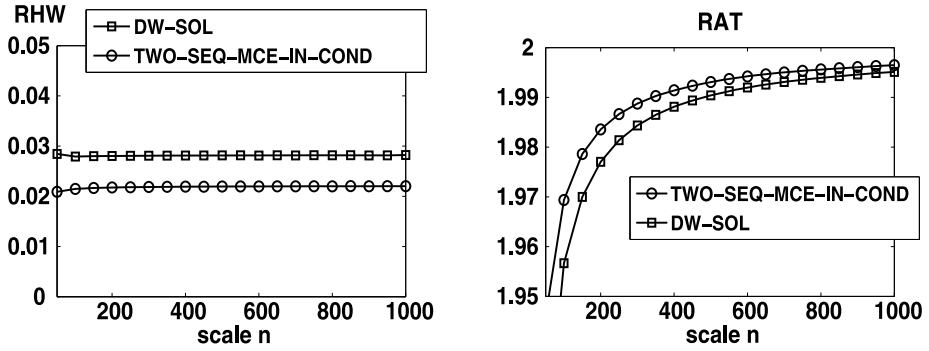

Fig. 8 Comparison of the DW-SOL algorithm and the best SEQ-MCE algorithm on the two-tailed problem with Laplacian jumps

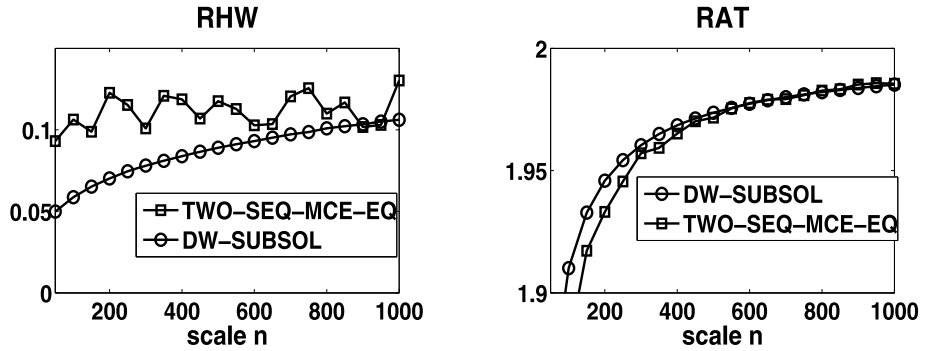

Fig. 9 Comparison of the SEQ-MCE algorithm with forced tilting and the DW-SUBSOL algorithm on the two-tailed problem with Laplacian jumps. (The STATIC estimator showed the same performance as the DW-SUBSOL)

Table 2 Exponential growth rate of the performance EFF of all estimators for the two-tailed problems at $n = 1000$, i.e. $\frac{1}{n} \log \mathrm{EFF}$

	Gaussian	Laplacian
TWO-SEQ-MCE-IN-COND	0.2650	0.4673
TWO-SEQ-MCE-EQ	0.2613	0.4615
TWO-STATIC	0.2630	0.4645
DW-SOL	0.2621	0.4665
DW-SUBSOL	0.2586	0.4628

whereas Dupuis and Wang (2004) proved only logarithmic properties. Again we see that our MCE algorithm with optional tilting and conditional last jump has the best performance.

5 Summary & conclusions

In this paper, we presented a way to obtain state- and time-dependent important sampling schemes by solving a sequence of minimum cross-entropy programs such as (1). When

the minimum cross-entropy programs contain inequality constraints, a consequence of our approach is that those aspects of the resulting change of measure are 'turned off' when no longer expected to be required for the remainder of the simulation. This gives some justification to the natural heuristic of 'turning off' the change of measure when it is no longer required. The basic idea of using MCE in this way was motivated by the recent state-dependent algorithms inspired by the large deviations approach (Dupuis and Wang 2004, 2007; Blanchet and Glynn 2006; L'Ecuyer et al. 2008).

Our technique, with a natural inequality constraint, was used to obtain a state- and time-dependent importance sampling scheme for estimating one-tailed probabilities of i.i.d. sums in which the jumps are light-tailed in Sect. 3. The solution to the associated MCE program (11) consists of a product of independent exponentially-tilted jumps distributions of the form (12). The state- and time-dependence is through the tilting parameter (14), and is 'turned off' when not required. Given the well-known connection between solutions to MCE programs and exponential tilting, it is no surprise that the algorithms obtained here are much the same as existing large deviations inspired state-dependent algorithms. In Sect. 3.1 we showed that the resulting algorithm is logarithmically efficient in general, and in Sect. 3.2 it was proven to be strongly efficient when the jumps are Gaussian.

Simulation experiments presented in Sects. 3.3 and 3.4 compared our algorithm and variants thereof to the classic optimal state-independent exponential tilting algorithm and a state- and time-dependent algorithm suggested in Blanchet and Glynn (2006); L'Ecuyer et al. (2008). Of all the algorithms, SEQ-MCE-IN-COND performed the best in terms of the usual performance measures of relative half-width RHW and the logarithmic efficiency ratio RAT. The worst performing algorithm, as one might expect, was the classic state-independent STATIC. In terms of the metric EFF, which incorporates CPU time, all of the algorithms were roughly equal. However, one should be aware that small differences in EFF can account for large CPU discrepancies in achieving the same variance level. This is especially so for large runs.

After considering the one-tailed problem, we considered the analogous two-tailed problem in Sect. 4. Therein, we proposed a mixed importance sampling estimator for this problem. Using results on mixed estimators presented in the preliminary Sect. 2.2, we directly obtained logarithmic efficiency for our estimator in general, and strong efficiency when the i.i.d. jumps are Gaussian. For our estimators, we used mixing probabilities found via MCE in Sect. 4.1.

Our simulation experiments for the two-tailed problem, given in Sect. 4.2, compared two algorithms based on subsolutions and solutions of an appropriate Isaacs equation (Dupuis and Wang 2004, 2007), a mixture of the classic state-independent estimator, and a single MCE algorithm TWO-SEQ-MCE-IN-COND. The MCE algorithm was formed using a mixture of two estimators (one for each tail), with each using the best performing estimator of the SEQ-MCE-IN-COND algorithm for the one-tailed problem. In these experiments, the sequential MCE algorithm was once again best performing on the performance measure RHW and RAT.

We have presented a method to obtain state- and time-dependent importance sampling estimators by repeatedly solving an MCE program as the simulation progresses. This MCE-based approach lends a foundation to the natural notion to stop changing the measure when it is no longer needed. We have used this method to obtain a state- and time-dependent estimator for the one-tailed probability of a light-tailed i.i.d. sum that is logarithmically efficient in general and strongly efficient when the jumps are Gaussian. We go on to construct an estimator for the two-tailed problem which is shown to be similarly efficient. From our simulation experiments, we conclude that slightly modified versions of our algorithms, in

which the last jump in the sum has the original distribution conditioned to make the associated event certain, performs marginally better than some other state- and time-dependent estimators in the literature (Dupuis and Wang 2004, 2007; Blanchet and Glynn 2006; L'Ecuyer et al. 2008).

Acknowledgements The authors would like to thank Dirk Kroese for funding a visit of the second author to the Vrije University Amsterdam during which much of the work was done. This was made possible by Australian Research Council support for grant DP0985177. Thomas Taimre further acknowledges the support of the Australian Research Council Centre of Excellence for Mathematics and Statistics of Complex Systems. The authors are grateful to two anonymous referees for their comments and suggestions.

Appendix

Let $\{(a_j(n))_{n=1,2,\ldots} : j = 1,\ldots,m\}$ be a collection of m infinite sequences of real numbers, and assume that each sequence has a finite limes superior, i.e.,

$$\limsup_{n\to\infty} a_j(n) < \infty.$$

Let for each n

$$j^*(n) = \arg\max\{a_j(n) : j = 1,\ldots,m\}.$$

(In case of a tie just choose the one with the lowest index.) In this way we have constructed a new sequence $(b(n))_{n=1,2,\ldots}$ with

$$b(n) = a_{j^*(n)}(n) = \max_{j=1,\ldots,m} a_j(n).$$

Then for any $j = 1,\ldots,m$ we get

$$b(n) \geq a_j(n) \text{for all } n \quad\Longrightarrow\quad \limsup_{n\to\infty} b(n) \geq \limsup_{n\to\infty} a_j(n).$$

Since this holds for all j, we have

$$\limsup_{n\to\infty} \max_{j=1,\ldots,m} a_j(n) = \limsup_{n\to\infty} b(n) \geq \max_{j=1,\ldots,m} \limsup_{n\to\infty} a_j(n).$$

For the reversed inequality, we reason as follows. Let $b^* = \limsup_{n\to\infty} b(n)$. Thus there is an infinite subsequence of indices $(n_k)_{k=1}^{\infty}$ such that $b(n_k) \to b^*$ as $k \to \infty$. The associated sequence $(j^*(n_k))_{k=1}^{\infty}$ is an infinite sequence of the numbers $j = 1, 2, \ldots, m$, thus at least one of these numbers occurs infinitely often, say $\tilde{j}$. In other words, there is an infinite subsequence of indices $(n_{k_\ell})_{\ell=1}^{\infty}$ of $(n_k)_{k=1}^{\infty}$ such that $b(n_{k_\ell}) = a_{\tilde{j}}(n_{k_\ell})$. This gives

$$\lim_{k\to\infty} b(n_k) = b^* \quad\Longrightarrow\quad \lim_{\ell\to\infty} a_{\tilde{j}}(n_{k_\ell}) = b^*.$$

And finally,

$$\begin{aligned}\limsup_{n\to\infty} \max_{j=1,\ldots,m} a_j(n) = b^* &= \lim_{\ell\to\infty} a_{\tilde{j}}(n_{k_\ell}) \\ &\leq \limsup_{n\to\infty} a_{\tilde{j}}(n) \leq \max_{j=1,\ldots,m} \limsup_{n\to\infty} a_j(n).\end{aligned}$$

References

Blanchet, J., & Glynn, P. (2006). Strongly efficient estimators for light-tailed sums. In L. Lenzini & R. Cruz (Eds.), *Proceedings of the 1st international conference on performance evaluation methodologies and tools*. New York: ACM.

Botev, Z. I., Kroese, D. P., & Taimre, T. (2007). Generalized cross-entropy methods with applications to rare-event simulation and optimization. *Simulation: Transactions of the Society for Modeling and Simulation International*, *83*, 785–806.

Bucklew, J. A. (2004). *Introduction to rare-event simulation*. New York: Springer.

de Boer, P. T. (2006). Analysis of state-dependent importance sampling measures for the two-node tandem queue. *ACM Transactions on Modeling Computer Simulation*, *16*, 225–250.

Dembo, A., & Zeitouni, O. (1998). *Large deviations techniques and applications* (2nd ed.). New York: Springer.

Dupuis, P., & Wang, H. (2004). Importance sampling, large deviations, and differential games. *Stochastics and Stochastic Reports*, *76*, 481–508.

Dupuis, P., & Wang, H. (2007). Subsolutions of an Isaacs equation and efficient schemes for importance sampling. *Mathematics of Operations Research*, *32*, 723–757.

Ethier, S. N., & Kurtz, T. G. (1986). *Markov processes: characterization and convergence*. New York: Wiley.

Glassermann, P., & Wang, Y. (1997). Counterexamples in importance sampling for large deviations probabilities. *Annals of Applied Probability*, *7*, 731–746.

Heidelberger, P. (1995). Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modelling and Computer Simulation*, *5*, 43–85.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, *106*, 620–630.

Jaynes, E. T. (1963). Information theory and statistical mechanics. In K. Ford (Ed.), *Statistical physics* (pp. 181–218). New York: Benjamin.

Kullback, S., & Khairat, M. A. (1966). A note on minimum discrimination information. *Annals of Mathematical Statistics*, *37*, 279–280.

L'Ecuyer, P., Blanchet, J. H., Tuffin, B., & Glynn, P. W. (2008). Asymptotic robustness of estimators in rare-event simulation. *ACM Transactions on Modeling and Computer Simulation* (to appear).

Ridder, A., & Rubinstein, R. Y. (2007). Minimum cross-entropy methods for rare-event simulation. *Simulation: Transactions of the Society for Modeling and Simulation International*, *83*, 769–784.

Rubinstein, R. Y. (2005). A stochastic minimum cross-entropy method for combinatorial optimization and rare-event estimation. *Methodology and Computing in Applied Probability*, *7*, 5–50.

Rubinstein, R. Y., & Kroese, D. P. (2008). *Simulation and the Monte Carlo method* (2nd ed.). New York: Wiley.

Sadowsky, J. S., & Bucklew, J. A. (1990). On large deviations theory and asymptotically efficient Monte Carlo estimation. *IEEE Transactions on Information Theory*, *36*, 579–588.

Smith, P. J. (2001). Underestimation of rare event probabilities in importance sampling simulations. *Simulation: Transactions of the Society for Modeling and Simulation International*, *76*, 140–150.

Tijms, H. C. (2003). *A first course in stochastic models*. New York: Wiley.